QL Today

Volume 7
Issue 2
July/August
2002

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

THE QL TODAY SUMMER MIX

SMSQ/E

Assembler

C

Shows

EasyPTR

QPTR

BASIC

Contents

Advertisers

in alphabetical order

QL *Today*

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
Mobile +49 1701 222277
email: JMerz@j-m-s.com
email: QLToday@j-m-s.com

English office:

Q Branch
20 Locks Hill
Portslade
BN41 2LB
United Kingdom

Tel. +44 1273 386030
Mobile +44 7836 745501
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk
email: QLToday@j-m-s.com

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk
email: QLToday@J-M-S.com

Co-Editor:

Bruce Nicholls
38 Derham Gardens
Upminster
Essex RM14 3HA
United Kingdom

Tel +44 1708 510764
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL *Today* is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL *Today*** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

QL *Today* reserves the right to publish or not publish any material submitted. Under no circumstances will **QL *Today*** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL *Today***. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

Editorial

Dilwyn Jones

At one point I thought this issue was going to be pretty empty as we were very short of material, but several people responded quickly to our help requests – thank you very much everyone.

I had hoped to bring you a Q60 review in this issue, as I'd had a machine on review loan from D&D systems for a while. Sadly, I failed to get the review done on time as the machine had to go back before I'd finished the review and my employers decided to make me work long hours and moved hospitals to start a new project which has proved less than straightforward. It's also meant I had less input to this issue of QL Today than I would have liked, and grateful thanks go to Jochen and Bruce for the extra work they put in.

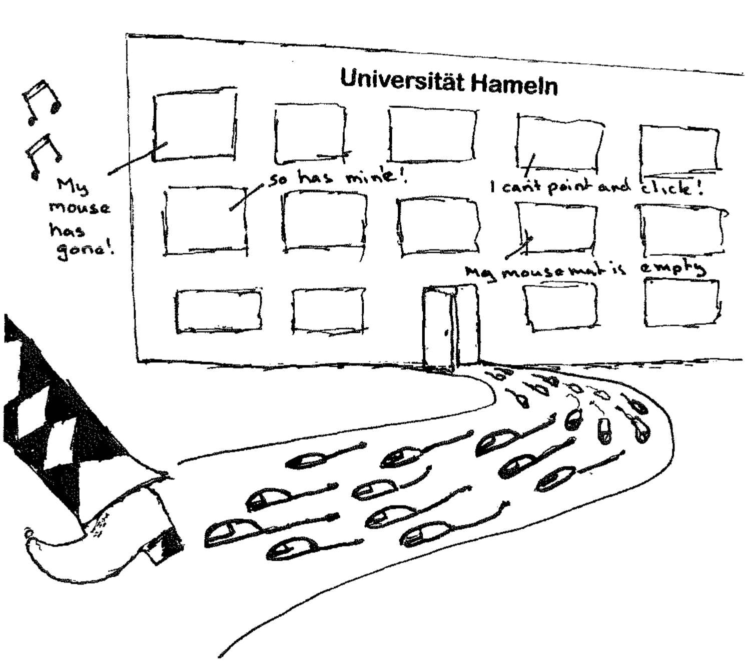
Wolfgang Lenerz has moved quickly to get the SMSQ/E sources out to interested programmers, and has produced a style-guide for those who wish to contribute to the project. While SMSQ/E is definitely not open-source in the same sense as Linux, for example, it's an interesting mix of the commercial and open worlds, where programmers can get the sources and contribute new modules or ideas to the project, all controlled by a "registrar", the official distribution is still a commercial project which will mean we will get the good backup service we have been used to from J.M.S. and QBranch, for example.
I have finally released the second version of the QL Documentation CD. This was a bit of a pet project of mine, getting as much QL documentation out there, freely available, as I could. Some of it is on my website, limited by space, and the rest available through either my PD library service or on a single CD-ROM. Further contributions to it are welcome!

A special thanks to our long running series authors Herb Schaaf and Norman Dunbar. Herb's graphics series reaches an incredible 29th part – just where does he get the ideas from? And if Norman has anything to do with it, we should by now all be machine code experts! Thank you chaps. And welcome too to a new contributor, Marcel Flipse from The Netherlands with an interesting hardware article.

When I was raving about QPC in these pages and elsewhere, I was accused of favouritism. Lately, just because I've had a Q60 on my desk, I've been enthusing about that too, so then I got several emails accusing me of bias towards Q60. Seems I couldn't win, but it did at least create much debate, which is always a healthy thing! To set the record straight: I think both are magnificent products in their separate ways and we should all be very grateful to Marcel Kilgus (author of QPC2) and Peter Graf (designer of the Q60) and those traders who continue to bring us such valuable products.

Cartoon

9.00am Monday morning. The students and staff at Hamelin University wake up and realise.......... The Piper is Back !



NEWS

Computer One Pascal

Jean-Yves Rouffiac has obtained permission from Paul Ives, former MD of Computer One (a company which published several QL programs in the 1980s) to distribute copies of Computer One Pascal via his website. The software remains the property of the original copyright holders. Computer One Pascal is a nice implementation of the Pascal programming language for the Sinclair QL. It features a full integrated development environment for coding, compiling and testing, and the ability to create stand alone executables. It was first released in 1984/5. Whilst not fully ISO compliant, it comes close, with only a few small differences. A number of extra procedures and functions are provided to allow you to make use of QL facilities from the Pascal system.

Computer One Pascal may be downloaded from Jean-Yves Rouffiac's site on
http://westhaven.uklinux.net/qwertyb/

Another Computer One product, their machine code monitor, has also been made available. This may be downloaded from
http://members.tripod.com/phpr/qhpc1mi.html

Both programs may also be obtained on disk from my PD library service.

A Tenner for Your Autograph!

Just Words! has now released AUTO-GRAPH, one of the most unusual programs ever written for the QL.

AUTO-GRAPH enables you to analyse most handwriting samples to form a profile of the writer. To use the program you need no previous knowledge or experience of graphology. All you have to do is follow the on-screen questionnaire. However, you will need a little learning time to familiarise yourself with different handwriting styles.

Graphology is a controversial subject, but is taken seriously by many companies when selecting job applicants. AUTO-GRAPH was written from an agnostic point of view and allows you to judge for yourself whether graphology is a science, a pseudo-science or just plain nonsense.

AUTO-GRAPH costs £10 or 15 Euros and is available from JUST WORDS! or QBRANCH. It is a pointer driven program and requires Toolkit 2 and either SMSQ/E or the pointer environment files to be installed on your machine.

QL Today has a review copy of this program waiting for a reviewer who'd like a go at reviewing this "unusual" program (to quote the author). If this subject interests you, please get in touch with the editor to offer your services a a reviewer!

Geoff Wicks, 28 Ravensdale, Basildon, Essex, SS16 5HU, United Kingdom. Tel: 02168 - 281826
E-Mail: **geoffwicks@hotmail.com** Web:
http://members.tripod.co.uk/geoffwicks/justwords.htm

Second Irish QL Show

The 2nd Irish QL show will take place this Summer - over the August Bank Holiday weekend, 24th to 26th August, 2002.

The dates are definite - just the venue and times etc. to be finalised (I have several options available to me).

It will most probably be on the Saturday, (this sound OK?) which would allow Sunday and Monday to hit the huge selection of Irish pubs in the Area ;-))

Or, if not into that, the huge amount of tourist attractions in the Area (Glendalough, just down the road, is the 2nd biggest tourist area in Ireland)

The venue will most likely be in Roundwood, about 6 or 7 miles away from the location of the previous one a few years ago. It will be near the IT and Internet Shop I own so I will be providing free internet access for the day too.

I will also organise a meal afterwards, and possibly on the Friday night also.

Please contact me for further details! email: **darrenb@esatlink.com**

Floating Point Unit Extensions and Q60

Claus Graf writes:

George Gwilt once told me that there is a extension that lets one use the Floating Point Unit of Q40 and Q60 (also QXL with replacement processor with FPU) from BASIC. The software is written by Simon N Goodwin and he was so kind to pass it on to me for release. There may be some people who already know it, but to me it was new and I also couldn't find it at the usual places for QL software.

It is free and open source. Please get it from **http://q40.de** (Programs section)

Speed increase can be huge, depending on the math functions used.

Wordscheck

This classic word count program is now available in an updated form, compiled with a current version of Turbo. Thanks to an idea from Paul Merdinian, this program can now extract a list of all the different words in a Quill or plain text file. Previously it would create a word count list; now you can use it to extract just word lists which would help with the creation of an index or a specialist word list extracted from a technical document for example.

The new version 1.8 may be downloaded from **www.soft.net.uk/dj/software/freeware/freeware.html**

BaseCon and Character Pick

I have just added two new pointer driven programs to the My Freeware page on my website.

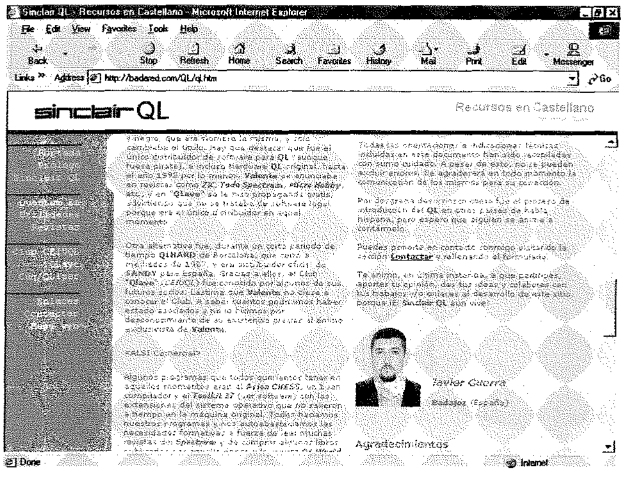
BaseCon - Pointer driven number base conversion, between binary/decimal/hex.

Character Pick - Pointer driven character selection menu, left click on a character from the QL character set to see its CODE value, right click to send it to another program via the stuffer buffer. Avoids frequent reference to the QL manual to find the keypresses needed for accented characters etc.

Both are about 30KB downloads from **www.soft.net.uk/dj/freeware/freeware.html**

New Spanish QL Site

QL Today has been informed of a new Spanish site dedicated to the QL, by Javier Guerra. Very little information was supplied to us, except that the site is in the Spanish language and the site address is: **http://badared.com/QL**



QL Documentation CD

Release 2 of the QL Documentation CD is now available. A considerable number of documents and diagrams have been added, contributed by many people since the original was released. Examples of files added include the QL Service Manual, several useful QL Today articles and most of the manuals for the TF Services hardware products have been added. The full list of items on the disk, a file called README_TXT on the CD, alone runs to some 55KB! The CD is supplied in ISO-9660 QXL.WIN format so should be usable on most QL emulators and Qubide systems using Thierry Godefroy's Atapi/CD thing. In addition, zipped copies of the various files are also available, so the CD can be used from non-QDOS systems to some extent too.

This CD is available direct from Dilwyn Jones and from Darren Branagh of Q-Celt Computing.

CDs available in the USA

American QLers can now order the Dilwyn Jones range of QL CD-ROMs from Phoebus Dokos in Indiana to save on Sterling currency conversion costs.

Phoebus R. Dokos, 941 Lilac Street Apt. #1
Indiana, PA 15701-3340, USA
Tel: +1 (724) 464 0199

PD Library grows

The Dilwyn Jones PD library service has grown to over 70 disks of general PD software, 15 disks of QL games, nearly 50 disks of QL screens clipart and over 25 disks of demo versions of commercial QL software. And the library is still growing, with a large numbers of disks of software yet to be sorted!

The catalogue is available by email, on my website, or as a text file on floppy disk.

Literature CD

Mike Edwards is working on a CD-ROM of classic literature for the QL. Available as text file, this should make an ideal source of reading material for those long winter nights! More details in the next issue!

The EURO and the Post

Although the EURO is valid currency and all stamps in the participating countries are priced in EUROs now, it is, unfortunately, not possible to use, say Austrian EURO stamps in another country but Austria. Please do not send your stamps around and include it with an update disk for return postage. If you want to enclose return postage (for dealers or Wolfgang Lenerz's SMSQ/E source), you still have to go for International Reply Coupons, if source and destination countries are not the same.

JMS News

After "the License" for the SMSQ/E sources has finally been sorted out (see final version on page 15 in this issue), and the Styleguide connected with it specifies that QMAKE should be used to link and make it, I thought it would be nice to do a SMSQ/E license summer special price. See JMS ad on page 27 for details.

New versions of QD and QMAKE have been made available for registered customers in the JMS BBS. I have improved some stippled colours so that it looks better in Hi-Res colour mode. The updates are not essential, and as usual can be obtained as well by sending in the master disks and IRCs ... please no foreign stamps (see note on previous page).

The BBS has been refreshed again. I would like to ask to use only the phone number 502014 in the future. Both 13 and 14 now end at the same modem, but I wish to free the 13 eventually. I had to change the ISDN exchange to a new one with internal ISDN bus, but fewer analogue ports, so I can connect only one modem. Not a problem, the traffic on the BBS is fairly low for the last couple of years, and the 502014 connects to the faster modem.

Super/EtherIDE

from Nasta

The goal: create a Qubide successor. This should update the Qubide circuitry with the latest ATA specs in mind, and also upgrade it with new functions. Here are the results:

The new IDE board comes in two flavors: SuperIDE, which is an IDE only interface, and EtherIDE, which is the same as the above, with a 10BaseT Ethernet interface built-in (more about this later). Both will be referred to as EtherIDE.

The board is a 4-layer one, exactly the same size as the Aurora. Indeed, it has mounting holes in the same places as the Aurora, so it will be possible to easily stack these two using appropriate spacers. There is no through-connector, the board is designed to operate on a backplane, powered by a 5V power supply. Also, the board is designed to work only on GC and later systems, and is of course 'forward compatible' with GoldFire. Although these seem like drastic design decisions, they have good reasons behind them, which will become apparent through the list of features.

Since hard drive technology has advanced considerably in the years following the creation of the original Qubide, the Super/EtherIDE takes full advantage of this with the following features:

Fully compatible with Qubide

I am trying to get ahold of Phil Borman, the author of Qubide, so that the code can be made open, or at least accessible. This would permit modifications to support the new functions of SuperIDE. In the interim, either version of Qubide code will work, but some features will not be as elegantly accessible as they would be in a fully fledged Qubide upgrade.

3 IDE channels for a total of up to 6 devices

- The 3 channels, A, B and C can be assigned using optional (*) jumpers so that the resulting order of devices and the boot device can be customized.
- For each channel, one of the devices (master or slave) can be an on-board device, see below.
- Channel C implements hot-swapping, the baility to insert and remove devices while the system is powered up. This ability will need support in software, the hardware default is emulation of a regular IDE channel, i.e. hot-swap is disabled.

Up to 3 On-board devices

- SuperIDE can accomodate one 2.5" hard disc (on channel A), and up to two compact flash (CF) cards (channels B and C) right on the board. The CF connector for channel C is placed at the edge of the board where it can be made accessible when the system is mounted in an appropriate enclosure, so the user can insert and remove it while the system is running, using the hot-swap function.
- When an on-board device is present, the standard connector for that channel can still connect to one off-board device.
- Use of on-board devices eliminates all cables to them, they simply slot into the connectors provided, power is provided from the backplane. This makes it possible to build very small systems, or use exclusively CF cards for minimum power consumption, and is ideal for something like the MinisQL.

Transparent ROM mode, simplified setup

- This special mode of operation enables the SuperIDE to operate without using up any ROM space. It requires a small piece of loading software and uses the RESPR version of the driver, stored in the on-board Flash ROM (see below). Because of this, it is possible to operate SuperIDE concurrently with RomDisq even on Goldcard systems.
- Because of the transparent mode ROM feature, there is only one address jumper and it is optional (*). It selects between GC and SGC/GF

systems in cases where the transparent mode is not used for some reason.

Simple software upgrades

- SuperIDE uses a 1/2 MB Flash ROM to hold the driver. A mechanism is provided to access the whole capacity of the Flash chip so that additional software can be loaded in it. Software upgrades can now be performed by the users themselves.

Odds and ends

- Unlike Qubide, which was created before the time these things became part of the official IDE specs, all signal lines are terminated for best signal integrity. Power for the IDE circuitry, onboard IDE devices, and in case of the EtherIDE, the Ethernet interface, is separately filtered to prevent mutual interference.
- Forward compatibility with GoldFire also insures that faster accesses will be possible to the hard drives, however the speed is still limited by 8-bit bus interfacin and the signal integrity on the bus. Even so, at least a 2x improvement in speed can be expected with GF.

(*) Optional jumpers and connectors

Because all of these add to cost but not necessairly to function in a given system, several connectors and jumpers are optional. It is left to the user to decide which ones their SuperIDE gets fitted with, at order time. These are:

- Channel assignment jumpers. Without them channels default to the order A, B, C - win1_ being the first device on channel A etc.
- ROM address jumper. Without it, the default ROM location is 0C000h, i.e. the ROM port address. Transparent ROM mode works regardless of the ROM location.
- CF and 2.5" drive connectors. User specifies what they want to have as these are quite expensive.
- Extra LED connectors. These are normally usefull only for developement and enable a separate LED activity indicator for each IDE channel. In the case of EtherIDE, a full complement of 4 Ethernet status LEDs is also optional, again, most usefull for developement. The board supports use of all options simultaneously when maximum flexibility is required.

EtherIDE speciffics:

The Ethernet portion of EtherIDE is equivalent to the Ethernet interface on the GF and is provided as an option in hopes it will create a developer platform, so that some form of Ethernet support is already available when the GF appears on the market. It is a 10BaseT interface, meaning its 10 megabit and uses a twisted pair medium. It should be noted that there is no software for this as of yet. However, having working hardware presents a huge potential for TCP/IP stack developement as it enables an already established system, such as a Linux or Windows box to be used as a 'simulator' for the rest of the network (for instance a web host or network sniffer) to help with testing and debugging. I hope this will advance both Ethernet and TCP/IP developement for the QL. The Ethernet hardware is highly integrated and consists of a single chip, a 10BaseT coupling transformer, and a few passive components. SuperIDE boards are the same as EtherIDE, the Ethernet speciffic parts are just not soldered on. This means that SuperIDE boards can be upgraded to EtherIDE. The hardware also shares SuperIDE's transparent ROM technology and operates without using up the limited address resources of the QL. The RJ45 Ethernet connector and the network activity LED connector are located along the edge of the board similar to the hot-swap capable CF connector. The single LED is 'intensity coded', it lights dimly when a proper link is achieved with another system or a hub/router, and lights brightly whenever data is being transmitted or received.

Technology:

Advances in technology have made it possible to put the entire IDE interface logic into a single programmable chip. SuperIDE requires only 3 extra chips - the Flash ROM, and two switch chips for hot-swap capability. EtherIDE adds one more chip for the Ethernet.

Status:

Project done, trying to gather up funds for a prototype run and then for full production.

Why this and not GF, which is what I should really be doing and what everyone is waiting for?

Well, GF requires money and a user base to exist when it finally gets done. EtherIDE did not take very long to develop, mostly because much of it is nearly verbatim Qubide circuitry. SuperIDE is a much smaller (though still sizeable) investment, and I am hoping to use it to rise money to get the GF to the market. In simple terms: I had to chose between possibly nothing and EtherIDE and GF - so I chose the latter.

Gee Graphics! (on the QL?) - Part 29

H. L. Schaaf

"And Even More Connecting the dots"......

This time we add the Gabriel graph, another subset of the Delaunay triangulation to the menu.

If we consider any Delaunay edge as the diameter of a circle, and there are no points inside that circle, then that edge is part of the Gabriel graph.

Continuing on from GG#28, take the mixture of code51_bas, D2V_bas, and the PROcedures for angl_frm, dist_btwn, SWAP and CYC that we had, with the listing "AddHull_bas" merged in.

Un-REMark lines 5830 and 5975 that refer to the Gabriel graph.

Now merge in the listing 'AddGabriel_bas'.

```
11000 REMark AddGabriel_bas, to go with GG#29
11010 REMark H L Schaaf July 2, 2002
11020 DEFine PROCedure Gabriel
11030 LOCal i,j,x1,y1,x2,y2
11040 FOR i = 1 TO DIMN(Dedg)
11050  test_val = 0
11060  mid_pt_x = (P(Dedg(i,1),1)+ P(Dedg(i,2),1)) /2
11070  mid_pt_y = (P(Dedg(i,1),2)+ P(Dedg(i,2),2)) /2
11080  x1 = P(Dedg(i,1),1) : y1 = P(Dedg(i,1),2)
11090  x2 = P(Dedg(i,2),1) : y2 = P(Dedg(i,2),2)
11100  test_rad = (dist_btwn (x1,y1,x2,y2))/2
11110  FOR j = 1 TO DIMN(P)
11120  IF ((j <> Dedg(i,1)) AND (j <> Dedg(i,2))) THEN
11130   test_dis = dist_btwn(mid_pt_x,mid_pt_y,P(j,1),P(j,2))
11140   IF test_dis >test_rad THEN
11150    test_val = i
11160   ELSE
11170    test_val = 0
11180    EXIT j
11190   END IF
11200  END IF
11210  END FOR j
11220  IF test_val = i : show_edge(i)
11230 END FOR i
11240 END DEFine Gabriel
11250 REMark end of listing AddGabriel_bas
```

I've mucked about and made some progress with Steve Poole's Voronoi program but still have some problems. Degeneracies?

Next time? You might well guess that Minimum Spanning Tree will come next. After that we can add a few more subsets of "neighbors" to the menu. Then perhaps we can go into some examples of degeneracies.

Footnote: Ruben Gabriel is a prominent statistician and emeritus professor of mathematics at the University of Rochester. He studied and wrote about geographic partitions, regions, and contiguity, connectedness and separateness.

We can see how these topics are related to Voronoi and Delaunay diagrams.

K. Ruben Gabriel & Robert R. Sokal "A new statistical approach to geographic variation analysis" *Systematic Zoology, Vol. 18, issue 3, pp.259-278. Sept. 1969*

[Editor's comment: Well, next time you're FIVE full years with us. Great, Herb, thanks in the name of the QL-Today team and all readers. And a quick word to the readers: try so Herb's creations in action, best explained by Herb himself. Always interesting at US shows, and maybe we have a chance in Berchtesgaden?]

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets.

Cost... £24

*************NEW**************

superHermes

A major hardware upgrade for the QL

All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/ keyclick) IBM AT kbd I/f // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software)	**£90** (£92/£93)
IBM AT UK layout Keyboard	**£11** (£13/£15)
Serial mouse	**£8** (£8.50/£9)
Capslock/scrollock LED	**£1** (£1.50/£1.50)
Keyboard or mouse lead	**£3** (£3.50/£3.50)
High speed serial (ser3) lead	**£4** (£4.50/£4.50)

Hermes available for £25 (£26/£27) **Working ser1/2 and independent input, debounced keyboard.**

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead)................... **£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if requd). Send disk plus SAE or two IRCs

MKI...£40 (£41/£43) **MKII...£65** (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq	**£39** (£40/£41)
4mbytes RomDisq	**£65**(£66/£67)
8 mbytes RomDisq	**£98** (£99/£100)
Aurora adaptor	**£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/ Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost .... £34 (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface **16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)**

2 amp (for 8 relays, small motors) .... **£40** (£43/£44)

4 amp total (for motors etc) .... **£45** (£48/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)...**£25** (£28/£29)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.........**£25** (£27/£28)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting.................£30 (£31/£32)

Temp probe (-40°C to +125°C)..............**£10** (£10.50/£11)

Connector for four temp probes..............£10 (£10.50/£11)

Data sheets...£2 (£2.50/£3)

Control software & manual (for all I/F).......£2 (£2.50/£3)

QL SPARES

Keyboard membrane	**no longer on sale**
1377 PAL	**£3** (£3.50/£4)
Circuit diagrams	**£3** (£3.50/£4)
68008 cpu or 8049 IPC	**£8** (£8.50/£9)
8301/8302 or JM ROM or serial lead	**£10** (£10.50/£11)
Power supply (sea mail overseas)	**£12** (£17/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address,/ postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details

10 May 01

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 Fax/BBS: 01442-828255
tony@firshman.demon.co.uk http://www.firshman.demon.co.uk

Comments to John Perry's review of Spelling Crib

Geoff Wicks

I can have no complaints about John Perry's positive review of Spelling-Crib, but there are a couple of points about which I would like to comment. I am also seeking some information over possible QTYP problems.

John writes that the Spelling-Crib manual warns that "apparently only one program at a time can use QTYP". This is not so. QTYP is strictly speaking not a spelling check program, but extensions that allow you to do spell checking. It is perfectly possible for programs to "share" QTYP. What is not possible is to have more than one dictionary loaded at any one time. Also only one of the programs can "control" QTYP. In practice this usually means that if you want to load a new dictionary, you can only do it using that program.

[Editor's comment: you can also load, save and add dictionaries through the QTYP Spell Extension (which is what you refer to) and the BASIC commands it installs.]

John asks why I use the forward slash "/" for a wildcard instead of the normal conventions of "*" for a string and "?" for a single letter. The simple answer is that is what QL-users wanted. One of the most common requests I had about Solvit-PLus was to allow "/" for wildcards as this is easier to type than "?" because no shift key is necessary. In Spelling-Crib it also had the advantage that "/" is by the alphabetic keys, whereas "*" is by the numeric keys. It speeds the typing of the search word.

Returning to QTYP. At the end of last year Dietrich Buder reported some problems with the QTYP_DED program that enables you to add words to or write dictionaries for QTYP. Although he was impressed by the compression of QTYP, he found problems with large dictionaries when working at the top of the dictionary in QTYP_DED. In particular he had crashes after saving and compression when he had added or corrected a word. Although the dictionary was usable for spell checking it could not be reloaded into the QTYP_DED, which gave error message of, I believe, an unsuitable file. It was however possible to expand this corrupted file and make a new dictionary from the text word list.

Some of my tests confirmed Dietrich's experiences, but unfortunately time pressures meant I was unable to do all the testing work I promised him. I have written large dictionaries for QTYP without countering similar problems, but then I use QTYP_DED only for building and compressing the dictionary and not for editing the word list.

I would be interested to hear the experiences of other people who have used QTYP_DED to write large dictionaries and, in particular, whether they included notes by some words, which I suspect may be associated with the problems Dietrich reported.

To Pay Or Not To Pay

Al Boehm

Roy Wood in his informative Byts of Wood (page 61 of the Mar/Apr 2002 QL Today) lamented the practice of "cracking" commercial software and then distributing the result for free. Indeed I have heard of freeware called Open Office that imitates MicroSoft Office including Word, Excell, and PowerPoint. I am a little wary of Open Office despite my judgmental scorn of MicroSoft for sharp business practice - two wrongs don't make a right.

On the other hand, it is legal under copyright law to analyze a program to find its algorithms and I/O format, and then write a program that does the same thing. You are not allowed to copy code to put into the new program. However, two people will often write the same code given a specific algorithm. So the trick is prove you wrote the code not copied it.

Open Office does provide source code. After determining the various formats that Word uses, a filter program could be written for the QL to read in Word documents.

A Specific Leson From the PC

Roy's specific example involved a freeware version of a 20 pound program to set up a ram disk on a PC. In IQLR Nov/Dec 1994 p 63, I showed how to make a ram disk for the PC! The PC ram disk was used in my QXL to gain additional fast storage. The instructions are not too arcane for most QL programmers and are included in the sidepanel for those QXL owners that may have need of such a thing.

To make a PC Ramdisk for the QXL

In the CONFIG.SYS file, anywhere after the HIMEM.SYS command, include the command

```
DEVICE=ramdrive.sys‹size›‹sector›‹directory›/E/A
```

where size is in kilobytes 4 to 32767 (64 is default), sector is 128, 256, or 512 only and is the number of bytes in a sector (512 is default), directory is the max number of directories, 2 to 64. E or A is used, not both. E means use extended memory; A means use expanded memory. If the program ramdrive.sys is in the C:\DOS\ directory then:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 1030 /E
```

will initialize a 1 megabyte ramdisk. Some PC OS used VDISK.SYS instead of ramdrive.sys.

On the QXL, the ramdisk shows up as another win drive and must be formated before use. This ramdisk format is very fast.

On the PC, a: and b: are assigned to floppy drives and the first win drive (QXL win1) is c:. The second win drive (QXL win2) is d: and may actually be a second hard drive or it could be the second partition of the first drive or a drive on the network, or the PC ramdisk. The third is e: and so on.

In any case, you will probably have to add to the CONFIG.SYS file:

```
LASTDRIVE=X
```

where X is a letter far enough along in the alphabet to allow for all various drives plus the ramdisk drive.

There doesn't seem to be much need of a PC ramdisk for other emulator users since they can use the better dynamic QL ramdisk. However, the PC ramdisk does seem to be a neat way for the QXL to communicate with say Q-emuLator or QPC2 so that both could be working on parts of a long number crunching program. . I have not been able to get ramdisk to work on QLAYW.

Therefore, the question becomes why did someone think they could charge 20 pounds for something that is free in the first place.

There are two answers. First I have not tested the PC ramdisk commands with the latest versions of Windows. Things may be (probably are) more complicated. The second reason is that if the commercial program has a good front end that is easily understood, it may be worth £20 to some people not to have to spend the time finding, understanding, and implementing the manual. This point has QL implications.

The QL Scenario

Forget about the PC. The first question is: Should we QLers "crack" commercial programs and give them away as freeware?

I had to answer this question recently. I have been writing and accumulating a collection of Ed Aids to help in editing S(uper)Basic programs. Some have been published in Quanta. Some of them use Qref/Qload which are commercial programs. I could, with some work, write them so they used freeware keywords instead of Qref/Qload. However, Qref/Qload are readily available and not that expensive. Plus I am way behind in my list of things I wish to program for the QL.

The second question is: Should we QLers go open source on our commercial products (The SMSQ/E issue!)

The sad fact is that no one expects to make a fortune writing QL software. Many of the QL commercial programs have been made freeware. Further, the lack of a money incentive has no doubt cut the number and quality of new QL programs.

We have a vast amount of QL freeware. Much of it is very good. The typical difference between the good freeware and the commercial products is in how "polished" the program is and in the documentation. Writing a program is fun; cleaning it up for diverse users and writing the documentation is work. Some kind of monetary incentive is requisite.

Grants as incentive

Much development in the US is not based on the hope of selling a product. Instead it comes from grants. NESQLUG has tried this approach. Suppose you as a programmer have a good idea for a program. You could write the program and try to sell it, or alternately you could petition NESQLUG or some other group for a grant. The amount from NESQLUG itself would be quite small.

However, in our experience, individuals wanted to get in on the ground floor and add to the seed money. Depending on the original agreement, the completed program could be made available as freeware for the betterment of the QL community, or it could be sold commercially with the original backers getting a discount. With the

small QL user base, it's worth considering.
Of course, I try to bolster those QL programmers that come out with a commercial product. I tell other people about them, write reviews about them, and when I have the money, I even buy them.

PS. My teenage exchange student guru informs me my definition of crack is old-fashioned. A "crack" is jargon for a program that defeats a program's copy protection. The "crack" is legal until it is used!

US QL Show - Washington, D.C. 1 June 2002

Alex Wells

At the end of May this year we were visiting the USA from Britain to attend a conference in Norfolk, VA, and to see our daughter in North Carolina. How could I resist a QL meeting nearby, when it was so perfectly timed?

The one day show was held at the Park Inn, just outside the Washington beltway, and actually in Maryland. Al Boehm (with Ruth Fegley, secretary of NESQLUG) had made the arrangements, complete with a collection service from local airfields for visiting QL-ers! This meant that short stay visitors could avoid the hassle of car rental. There was a Friday evening get-together in the local Chinese restaurant for those arriving in time, and a 'Maryland Crab Fest' on Sunday to add local flavor.

Al Boehm

Al and Dorothy had chosen a hotel which combined a metropolitan location with reasonable prices and accessible local amenities - shops, food and drink were 'just around the corner'. This was very handy for Europeans or Californians hoping for a no-driving booze on Friday and Saturday night.

Saturday started with checking in the 30 visitors - a number which included about 8 European residents, 4 californian visitors, NESQLUG 'locals', plus several family members and friends not active at the QL keyboard. Dorothy Boehm had generously offered to show wives and others around Washington on Saturday. I had planned to drop Pat and Fiona off at George Washington's house just across the river, so we were delighted when Dorothy offered to include it before starting the more conventional city tourist attractions - many thanks from all for a stimulating day out!

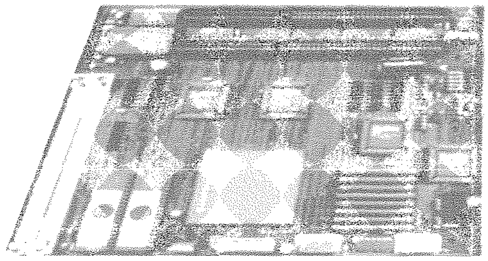
The morning session of the show was spent with our august body of traders (Jochen and Roy Wood) showing the QL flag and answering queries, but probably not doing enormous amounts of business. We are lucky to get this 'devotion to duty', as the costs must be far greater than the income. Marcel Kilgus was present to describe the updates to QPC2 and SMSQ/E for QPC - there were to be more improvements later in June. Darren Branagh and Steve Reyal had their camera going (as was the case at the Manchester AGM) so if you want to see many of the leading figures on the US QL scene, get the CD. Darren and Steve played their Quanta AGM CD in the afternoon session.

In addition to the conventional trading, Al Boehm had set up two 'virtual trade tables' for the mutual benefit of the 30 or so visitors and traders who could not be present. The first of these unmanned stands showed the D & D printed publicity material about the Q40i and Q60. A second more modest area was devoted to 'Bible Helps' bible study disks with Greek and Hebrew texts available - the QL was always adaptable to different

Herb Schaaf, Mary-Ellen Malloy, Bob Malloy, Jochen Merz, Roy Wood and Joe Lapunzina (left to right)

New Millennium.
New Century.
New QL.

What more could you want?



Features

- Q40i : 68040 CPU, 40 MHz, MMU, FPU
- Q60/60 & Q60/66: 68060 CPU, 60/66 MHz, MMU, FPU
- Q60/80: 68LC060 CPU, 80 MHz, MMU
- 68060 superscalar architecture, dual execution units
- Up to 160 BogoMIPS performance for QDOS+SMSQ/E
- 16 to 128 MB RAM, PS/2 module sockets
- 256 kB ROM (mainboard supports up to 1024 kB)
- Highspeed 32 bit graphics, plus original QL modes
- Up to 65536 colours at 1024 x 512 pixel resolution
- Multisync monitor output (15 pin HD connector)
- PC Keyboard interface (DIN)
- 20 kHz Stereo sound
- Battery buffered clock, 2 KB nonvolatile RAM
- Controller for 2 floppies and 2 IDE harddisks or CDROM
- 2 Serial ports with 115200 Baud, Parallel port (on I/O card supplied with mainboard)
- Hardware extension slot supports ISA cards
- Fits directly into AT Minitower or other standard case
- +5V / +12V power supply
- No tinkering, no parts from original QL needed
- Mainboard size 8.2 x 6.3 inch
- Can boot SMSQ/E in a few seconds, directly from ROM
- Three different operating systems available! SMSQ/E, QDOS Classic, Q60 Linux

Prices	
Mainboard	
Q40i	**£ 284.00**
Q60/60	**£ 390.00**
Q60/66	**£ 463.00**
Q60/80	**£ 614.00**
RAM	
16 MB**	**£ 17.00**
32 MB	**£ 34.00**
64 MB	**£ 36.00**
80 MB	**£ 53.00**
128 MB	**£ 72.00**
I/O Card (FLP,IDE,SER,PAR)**	**£ 14.00**
Operating System	
SMSQ/E for Q40i/Q60**	**£ 10.00**
QDOS Classic for Q40i/Q60*	**free**
Q60 Linux CD	**£ 15.00**
Ethernet Card 10 Mbit/s	**£ 17.00**
Cased assembled computers	
Type A (Mainboard in case)	**£ 40.00**
Type B (Fully built system with mainboard, 1.44 MB Floppy, 20 GB Harddisk, 54xCDROM)	**£ 152.00**
Preinstalled software package	
QPAC1, QPAC2, FiFi, QD, PROWESS and much more, over £100 worth	**£59.00**

** Standard parts, always required for operational mainboard *on mainboard support disks

Shipping and handling is extra. Prices may change due to semiconductor costs or exchange rates. All mainboards with VGA lead, sound adaptor, support disks and manuals. Please note: The Q60/80 is not available with floatingpoint coprocessor.

Q60

Q40

Website and technical information:

www.q40.de

Email: info@q40.de

D&D Systems

P.O. Box 5813, Ripley, Derbyshire, England DE5 9ZR
Tel. +44 (0)1773-740170, FAX +44 (0)1773-748399
Email: sales@q40.de

Financially assisted by a loan from QUANTA

Take the power back in your hands.

Darren Branagh and Steve Reya

fonts! A third stand contained a small quantity (by recent Manchester AGM standards) of bits and pieces. The main attraction was a collection of 12 old 275MB hard drives - ideal sizes for Qubide systems - rescued by Al. His innovative pricing policy (1 for $1, 2 for 50 cents) ensured that they all went in no time.

In the afternoon, we had informal presentations or demonstrations, including the AGM video. Herb Schaaf connected his separated QL components, described as 'the first distributed computing complex', and ran some of his 'Gee Graphics' programs. He gave a fascinating demo and answered many questions. It is good to see the QL performing original calculations rather than word-processing!

Nasta gave us an update on his hardware projects, and the GoldFire in particular always arouses special interest. The speed of the GoldFire will roughly equal that of the Q60/66, except for floating point since a 68EC060 CPU (without FPU) is used. It's speed is about 13x that of the SGC. As always, the financial aspects were the main cause for delays.

When Goldfire arrives, it is likely to release secondhand SGCs into the market. There was also a hint that some bargain new Auroras may appear soon. Aurora display drivers for SMSQ/E were also a topic of discussion.

'Nasta' again and Jeff Brown

Another possibility that Nasta discussed was building a 'portaQL' based on the 68SZ328 DragonBall CPU, used in the 68k based Palm PDAs. He said this is the highest frequency 68000 core in existance (as opposed to 68020+ cores). It works at 66MHz, is fully 68k compatible, has a built-in SDRAM controller, serial port, and USB. It might even provide high color VGA monitor or LCD output without needing too many additional components.

There was a discussion about color palettes with Jim Hunkins involved, trying to find methods that would work with ALL the platforms, QL, Aurora, QPC, Q40/60 and the upcoming GoldFire. It was great to see the forces that unify our interests at work, to make sure that everyone shares in the advances.

Marcel Kilgus, Zeljko Nastasic and Jim Hunkins

Jim Hunkins demonstated his QDT software project - this will be the QL's answer to 'Windows' (particularly for those who find the simple icons and windows of QPac2 a bit spartan!) when it is in a usable state. See the article by Jim in the May/June issue of QL-Today.

There were many other topics discussed which I either missed (like the evening NESQLUG meeting) or did not understand, but the afternoon session showed that there could still be exciting times ahead for the QL and its derivatives.

So, support your favorite computer and visit a QL show near you. If there is nothing around, why not set up your own local show?

Jochen adds: speaking for Roy and myself, and probably for the other visitors too, we would like to thank Al again for taking the trouble to prepare a show. Although it not even covers the air fares, we enjoyed it very much: the show, meeting all of you again, the dinner, we had a lot of fun going out the two nights (fighting the jetlag, which actually helped!). We decided to miss the crab fest, but we were very worried about missing the plane which we had to catch in the afternoon. Hope you find a date and a venue next year which will please everybody and allow as many people to come as you did this year.

And thanks to Joe Lapunzina for providing us with pictures to guide the articles - like he does every year.

The SMSQ/E Licence and distribution of source codes

Wolfgang Lenerz

I'm pleased to inform all of you that the final version of the licence is now out and that, as a consequence, it is now possible to obtain the source code from me.

The text of the licence is as follows:

LICENCE FOR SMSQ/E

This licence sets out the terms under which this software and the documentation may be used. Any use of this software implied adhesion to these conditions of use and this licence.
Headings used in this licence are there for indicative purposes only.

1 Definitions

The software The software is the set of source code, intermediary code, binary code, compiled code and documentation, or any part or parts thereof, making up the SMSQ/E operating system and all of its extensions as supplied herewith. The binary version, also called compiled version, of the software is the software in the compiled form of the source code.

Copyright holder The copyright holder is the person holding the copyright over the software as defined above. This person is Tony TEBBY.
Part of the software may be copyright by other persons. These persons are not the "copyright holder" as defined herein.

Registrar The registrar is the person appointed by the copyright holder to keep coherent versions of the software and issue official versions thereof from time to time. The person, duties and rights of the registrar are set out in an annex hereto.

Resellers A reseller is a person entitled by the copyright holder to sell the binary (i.e. compiled) code to the end users. A list of the current resellers is appended hereto.
Software author A software author is either Tony TEBBY or any person making a change/addition/modification to the software.

2 Copyright

The software is copyright (C) Tony TEBBY, 1990 to 2002.
Additional copyright may lie with other persons.
These persons are identified in an annex hereto.

3 Conditions of use

This software is supplied under the following conditions:

A - Source code.
In the following paragraphs, the term "source code" also encompasses documentation as it exists under this licence.

Obtaining the source code
Any person so interested may have access to the source code of the software. To obtain the source code, a request in this respect should be made to the registrar. Such request should be accompanied by 3 International Reply Coupons. The version of the source code thus distributed via the registrar, on

CDROM, is the "official version" of the source code. The registrar is entitled to distribute all or only part of the source code and may also distribute, at his convenience, the source codevia other media.

Distribution of source code by others than the registrar Any person may distribute the source code to others, provided however that the following conditions are adhered to by the person thus distributing the source:

- Such a distribution must be made entirely free of charge - no fees whatsoever, for copying or the media on which the software is copied or otherwise, may be levied.
  The distribution of the source code must contain a copy of this licence and a clear indication that this licence must be read and agreed upon by the recipient before using the source code.
- Such a distribution may only be made in either of two forms: Via a CDROM or via Email.

* Via CDROM

Exceptionally and only if distribution is made via CDROM, the person distributing the source code may request 3 IRCs and a blank CDROM from the recipient. All of the software, including the documentation and this licence must be distributed on the CDROM.

* Via Email

Distribution via Email is only allowed if the Email is sent out individually to the recipient having made the request. A system whereby an Email is automatically generated by software or hardware upon each or any request by a future recipient is NOT allowed. When distribution is made via Email, it is allowed to send only parts of the source code to the recipient, provided however, that in accordane to the requirement set out above, this licence is always sent together with such (part of the) source code.

- Any distribution of the source code by persons other than the registrar results in a version that is NOT an official version of the source code.

Compilation of source code Any person may compile the source code and thus obtain a binary (compiled) version and use it for itself. However, unless otherwise provided herein, binary code may NOT be distributed.

B - Changes/additions/modifications

Any person may make any change/addition/modification to the software, and distribute it as source code together with the source code of the software under the above conditions.

Any person having made such a change/addition/modification is a software author and, of course, retains entire copyright over the change/addition/modification thus made.

Such change/addition/modification may also be distributed in binary form, provided however that such a distribution does not contain any part of this software in binary form.

Submission of changes/additions/modifications to the registrar Any person having made a change/addition/modification to the original source code may propose this change/addition/modification to the registrar for inclusion in the official version.

When such a proposal is made, the person proposing it may state that the inclusion is to be made for free or against payment of a fee for each copy of SMSQ/E actually sold containing such a change/addition/modification.

When such a proposal is made, the person proposing it may state whether its change/addition/modification is to be:

- distributed in the official versions of the source and binary codes, or
- distributed in the official versions of the binary codes only, or
- not distributed in the official versions, but alongside them

.

If nothing of the sort is stipulated, then it is explicitly agreed that the inclusion in the official version is to be made for free, and that it will be distributed with the source code and under this here licence. Any code distributed in the official version (but not code only distributed alongside the official version) falls under this licence, to which the author explicitly agrees.

Authors having made a change/addition/modification retain copyright over their own code. The copyright will be acknowledged in an annex hereto.

By making such a proposal for inclusion, the person making the proposal explicitly states that the code is free from patent restrictions and it grants the registrar and the copyright holder the irrevocable right to include the change/addition/modification in any and all future versions of SMSQ/E, under the same conditions..

The fact of proposing such a change/addition/modification to the registrar for inclusion in the official version, whatever the form thereof, implies acceptance of these conditions.

The registrar is not obliged to include in official versions of the software any code submitted to him. However, consent to include it shall not be withheld unreasonably. The registrar is free to include code in one version of the software, but to exclude it from other future versions of the software. An annex may define the conditions under which the registrar may refuse code to be included.

Code submitted to the registrar must adhere to a style guide, as set out in an annex hereto or a separate document.

C - Binary code

Unless by prior agreement, binary code of the software may only be distributed by resellers.

The resellers may only distribute binary (compiled) versions of the official version of the software as obtained from the registrar.

The distribution of the binary versions of the software is made against payment by the reseller to the software author(s). Each reseller is free to set the prices he wishes for payment from his clients/customers, provi-

ded however, that for each new copy sold, an amount as determined in an annex hereto is paid to the software author(s).
Such a payment may be made by the reseller either directly to the software author(s) or to the registrar. The registrar is thus authorized to receive and collect payment, under the explicit condition however, that he MUST not take any payment himself, and distribute the entire money received to the software author(s).
Payment for binary versions of the software does not necessarily accrue on each distribution of binaries. Users having bought previous versions of SMSQ/E may be entitled, at the resellers' discretion, to free upgrades for certain versions of the binaries. This necessarily implies that the user obtaining an upgrade must have a previous legitimate copy of the SMSQ/E operating system. It is the reselers' responsibility to make sure that the person in question is entitled to a free upgrade.
However, the registrar, after consultation with the resellers, may require that any version of the binaries may not be distributed as a free upgrade, but that payment to the software author(s) MUST accrue on it.
Any person may petition Tony TEBBY or (preferably) the registrar to become a reseller. An authorization to become a reseller will not be unreasonably withheld. An annex hereto states the current resellers.
A reseller must supply support to the end user having bought SMSQ/E from him.

D - Test versions

As an exception to the prohibition of distribution of binary versions of the software other than through the resellers, you are hereby granted the right to distribute binary versions of the software to a maximum of 10 different persons (whatever the number and moment in time of the changes/additions/modifications you make), provided however that:
(a) you have made a change/addition/modification to the software compared to the official version and (b) the person receiving the software from you undertakes to destroy the binary version

- after 2 months of receipt thereof,
- as soon as you inform such person that the version is no longer a test version but a final version,
- as soon as any such change has been submitted to the registrar and accepted by him for inclusion in the official version whatever comes first.

The person receiving the binaries from you must also undertake not to distribute binary versions to anybody else - except yourself -, even if he/she did make any change/addition/modification to the code.
Even if you make several changes/additions/modifications to the software, you may only distribute test versions to the same 10 persons maximum.

E - Developer versions

The registrar may set up a system whereby identified software authors may have access to, and possibly exchange among themselves, source code or binary versions being actually developed.
End of licence

There are also the following annexes:

Annexes

Copyright

All of this software is copyright (C) Tony Tebby,1990 to 2002. Part of this software is copyright (C):

Cost of binaries.

The cost of the binaries, i.e. the money to be paid to software authors (either directly or through the registrar) is of 10 EUROS. This is paid to:
- Tony Tebby for 10 EUR

Resellers

The current resellers are:
Marcel Kilgus
Jochen Merz (JMS)
Roy Wood (Qbranch)

The registrar and inclusion of Code

The registrar is Wolfgang Lenerz.
6, rue Daunou
77340 Pontault-Combault France

The main work of the registrar is to make sure, as much as possible, that coherent versions of the source code exist for all machines on which SMSQ/E runs.
When code is submitted to him for inclusion in the official version, the registrar shall not withhold consent to include it unreasonably, but his decision in that respect is final. Code may only be excluded if there is a reason preventing it from being included.
If the registrar chooses not to include any code, then he MUST inform the software author of the reason(s) preventing him from including this code. The registrar explicitly agrees that these reasons may be made public by the software author, or others.
Reasons which may prevent code from being included in the official versions of the software include, but are not limited to:

- Non adherence to the style guide.
- Probability of the code being bugged.
- The fact that the code is useful only to one machine even though is could be made useful to more machines without too much effort.
- Clashes between the proposed code and existing code or code being written by others.
- The payment requested by the software author is inadequate in the registrar's opinion.

Style guide

The styleguide is contained in a separate document.

end of annexes

And finally, there is a styleguide, which reads as follows:

Official SMSQ/E style guide

Revision 1.0

1.0 Purpose

The purpose of this document is to keep a single coding style within the whole project. SMSQ/E has evolved over more than a decade with many people involved, therefore most but not all existing source files comply with the style described in this guide. New source files however have to comply. If changes in existing files are done that don't comply with the style described here it is your choice to either adapt to the style of the given file or make your changes in the style described in this guide.

2 Generic requirements

2.0 Development system

The standard distribution is assembled using the QMAC, QLINK and QMAKE assembler tools. All submissions must be compatible with these tools, the only exception being hardware dependant code which may need other tools for certain purposes (e.g. 68020+ assembler commands).

2.1 Assembler

All parts not specific to a certain hardware must be written in plain 68000 language and must be compatible with the QMAC assembler syntax.

2.2 Character set

The normal QDOS/SMSQ character set is to be used.

2.3 TAB stops

TAB limits need to be set to 8 characters. You are encouraged to configure your editor to compress multiple spaces to TAB characters in order to keep files small.

3 Assembler files

3.1 Generic file structure

A source file starts with one or more header lines followed by a changes list, the "section" command, xdefs, xrefs, include files and finally the code itself. Example:

```
; Routines to do something brilliant  V1.02  (C) 2002  Fred Flintstone
;                                                      Barney Rumble
; Additional information about the file (optional)
; 2002-01-01  1.00      First release (FF)
; 2002-06-20  1.01      Added br_evenbetter function (BR)
; 2002-12-31  1.02      Fixed serious buffer overflow in sub-function
;                       br_doanything (FF)

        section brilliant

        xdef    br_super
        xdef    br_evenbetter

        xref    cv_ctype

        include 'dev8_keys_sms_io'

        [Code]

        end
```

The header line gives a short explanation of the purpose of the file, the current version number and the list of names of people that hold the copyright (additional names are added in an extra line).

The changes list is something new and so far you won't encounter it in any original files of the distribution. Now that the code is available to more developers keeping a detailed track of the changes within a file becomes a must. When working on a file, always increase the version number and write down your changes in the list!

The format of the list is

```
; YYYY-MM-DD v.vvDescription of change (full or abreviated name of author)
```

The date is in ISO 8601 notation (big words for "year first, month next and day last).

3.2 Headers

A header within a source file always starts with the line

;+++

and ends with the line

;---

All routines available to external files should have a function header. Function headers contain a description of the function followed by a detailed in/out table as shown here:

```
;+++
; XYZ driver - read function (one line description, optional)
;
; This function does nothing really, it is just an example for a
; function
; header (detailed discussion of function)
;
;       d0 cr   drive number / error status
;       d1  r   byte read
;       d2    s
;       a3 c  p linkage block
;       a5 c  u pointer to something
;
;       status return standard
;---
```

The register list is a sorted list of all registers affected (D0 first, A6 last). Following the register name is a standard field with several characters that define the function and behaviour of the register: d0 xy z description (note the space between xy and z)
x: " " or "c" = "call parameter"
y: " " or "r" = "return parameter"
z: " ", "p" = "preserved", "u" = "updated" or "s" = "smashed"
So in the example header above d0 is a call parameter with the drive number and also a return parameter with the error status. D1 is a return parameter, too, whereas D2 is just smashed. a3 is a call parameter and is preserved. a5 is a call parameter that gets updated within the call.
Registers not listed must be preserved if not stated otherwise.
After the register field the status of the flags after the call is documented. This can be for example:
- "status return standard"
- "status return arbitrary"

3.3 Cases

Labels and the assembler mnemonics themselves are lower-case. Comments and headers are either lower or mixed case.

3.4 Comments

Comments are started using the ";" operator and must be written in English.

3.5 Labels

Labels are lower case, words are separated using the "_" character. Normally a label has its own line. The labels belonging to the body of a function usually start with a short hand for the function name, i.e.
com_check -› comc_loop
com_rxen-› crxe_iact
com_iserve-› comi_error ...

3.6 References to include and other files

All new references to include and other files must be made to files located on device 'dev8_'. The source code now contains reference to files located on drive "win1_". To allow some sort of more drive independence, it will be attempted to migrate these references, over time, to dev8_. Use dev_use 8,xxxy_ to set the dev device to whatever you want. Obviously, for the time being, this will be win1_, since all of the code still refers to this.

4.0 Final words

This documents was drafted by Marcel Kilgus. You can send comments and suggestions to
styleguide@mail.kilgus.net

Well, that's it already. In this text I tried to cover Tony Tebby's general programming style as acurately as possible (though it might not be complete yet). I personally have adopted his style for all my 68k related work. Nevertheless this text can only act as a guideline. In some special cases it might make more sense to write things differently. In that case, it might be a good idea to consult with the registrar, who will attempt to find a solution that all authors can live with.

So long, have fun with the sources.

end of styleguide

The styleguide is a living document, and may evelove from time to time. I hope that all of you can now enjoy the sources...

Why not DBF2DBAS for a change?

François Van Emelen

One of the major obstacles (at least for me) to go on using the QL on a daily basis is the lack of compatibility of data files with other systems. Data produced/created with QL software can't directly be used with MsDos/Windows applications and vice versa.

Fortunately some clever programmers have tried (with success) to narrow that gap. To name but a few: Geoff Wicks with QL2PC, Pedro Reino with

QUILLRTF (QL progs) and Phil Borman, Norman Dunbar and Dilwyn Jones with Stripper and Stripper2 (Windows prog).

But there is still a lot to do.

DBF2DBAS is my modest contribution to the QL community. Of course, it is a Sbasic program written by an amateur. It needs TK2 and menu_rext (Jochen Merz).

Feel free to use it, correct it and make it better. Once converted, the DBS-file can be used with your own Sbasic or C programs or with Wolfgang Uhlig 's excellent frontend 'Suqcess'. The DBAS Sbasic extensions can be found on Thierry Godefroy's website. (also available from Quanta library and the usual sources of good free software-editor)

Still on my wish-list:

- DBAS2DBF, DBF2QSPREAD, QSPREAD2DBF, QSPREAD2DBAS, etc to convert data from one application to another.
- A replacement for Easel
- A long string editor or Sbasic extension (could be handy to edit DBS string fields of 2048 char long)
- Sbasic functions to convert a Qdos string to Msdos/windows string (to keep the accents) eg. a$=QDOS2DOS(b$), a$=QDOS2WIN(b$), a$=DOS2QDOS(b$), a$=WIN2QDOS(b$)
  A way to print/display numeric values in a readable way; eg.
  ```
  myeuros=9999999999.99
  print_pict(myeuros,"000,000,000,000,000.00")
  ```
  would display/print 9,999,999,999.99 instead of something like 1E10)

I would like to thank David Denham for his articles 'Clocking on'

Database Structure (.DBF)

A database file is made up of a header record and data records. The header record defines the structure of the database and contains any other information related to the database. It starts at file position zero.

The data records* follow the header (in consecutive bytes) and contain the actual text of the fields. The length of a record (in bytes) is determined by summing the defined lengths of a1l fields.

Numbers in this file are represented in reverse bytes.

__Data File Header Record__
Bytes / Description

00 Type of data file:
FoxBASE+/dBASE 111 Plus, no memo - 0x03
FoxBASE+/dBASE 111 Plus; with memo - 0x83
FoxPro/dBASE IV, no memo - 0x03
FoxPro with memo - 0xF5
dBASE IV with memo - 0x8B

01-03 Last update (YYMMDD)
04-07 Number of records in file
08-09 Position offirst data record
10-11 Length of one data record (including delete flag)
12-31 Reserved
32-n Field subrecords**
n+1 Header record terminator (0x0D)

Field Subrecords***

Bytes / Description
00-10 Field name (maximum of 10 characters - if less than 10 it is padded with null character (0x00)

11 Data Type:
C - Character
N - Numeric
L - Logical
M - Memo
D - Date
F - Float
P - Picture

12-15 Displacement of field in record
16 Length of field (in bytes)
17 Number of decimal places
18-32 Reserved

Notes to Data File Structure

* The data in the data file starts at the position indicated in bytes 08-09 of the header record. Data records begin with a delete flag byte. If this byte is an ASCII space (0x20) the record is not deleted; if the first byte is an asterisk (0x2A) the record is deleted.
** The number of fields determines the number of field subrecords. There is one field subrecord for each field in the datbase.
*** See the System Capacities in this appendix for limitations on characters per record, maximum fields, etc.

System Capacities

	FoxPro	Foxpro Extended
Database and Index Files		
Maximum # of records per database file	1 billion*	1 billion*
Maximum # of characters per record	4,000	4,000
Maximum # of fields per record	255	255
Maximum # of databases open at one time	25	25
Maximum # of characters per database field	254	254
Maximum # of characters per index key (.IDX)	100	100
Maximum # of characters per index key (.CDX)	254	254
Maximum # of open index files per database	unlimited**	unlimited**
Maximum # of open indexes in all work areas	unlimited**	unlimited**
Field Characteristics		
Maximum size of character fields	254	254
Maximum size of numeric (and float) fields	20	20
Maximum number of characters in field names	10	10
Digits ofprecision in numeric computations	16	16
Memory Variables and Arrays		
Default # of memory variables	256	256
Maximum # of memory variables	3,600	65,000
Maximum # of arrays	3,600	65,000
Maximum # of elements per array	3,600	65,000
Program and Procedure Files		
Maximum # of lines in source program files	unlimited	unlimited
Maximum size of compiled program modules***	64K	64K
Maximum # of procedures per file	unlimited	unlimited

* The actual file size (in bytes) cannot exceed 2 gigabytes for single-user or exclusively opened multi-user .DBF files. Shared multi-user .DBF files with no indexes or .IDX indexes cannot exceed 1 gigabyte. Shared multi-user .DBF files with structural .CDX indexes cannot exceed 2 gigabytes.

** Limited by memory and available file hndles. .CDX files use only one file handle.

*** A program module is one procedure. A program or application can contain an unlimited number of program modules.

```
100 OPEN#0,con_: WINDOW#0,800,600,0,0 :INK#0,0:PAPER#0,7:CLS#0
110 OPEN#1,con_: WINDOW#1,800,600,0,0 :INK#1,0:PAPER#1,7:CLS#1
120 OPEN#2,con_: WINDOW#2,800,600,0,0 :INK#2,0:PAPER#2,7:CLS#2
130 OUTLN
140 AT#2,1,2:PRINT#2,'Convert Foxpro DBF files to DBAS files'
150 AT#2,2,2: INPUT#2,'Press Enter key to continue',ab$
160 REMark prepare arrays to extract data from DBF file
170 DIM ID_dbf_file$(1),DATE_update$(4),TOT_rec$(4),POS_1field$(2),LEN_dbf$(2),RES_1$(20),REC_detail$(32)
180 WHEN ERRor
190 erreur=ERNUM
200 IF erreur :REPORT_ERROR ERNUM,,,2 :PRINT ERNUM,ERLIN:PAUSE:QUIT
210 END WHEN
220 OPEN_DBF_FILE
230 OPEN_DBAS_FILE
240 WHEN ERRor
250 IF ERNUM:CLOSE#3:REPORT_ERROR ERNUM,,,2:PRINT ERNUM,ERLIN:PAUSE:QUIT:END IF
260 END WHEN
270 NAME_THE_FIELDS
280 REMark prepare array for field names and structure for DBAS
290 DIM nvv%(noMemo-1,1),newfield$(noMemo-1,11)
300 y=0
310 FOR x=0 TO numb_of_fields-2
320  IF  '-' INSTR name_of_field$(x+1)=0
```

CAMBRIDGE Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*

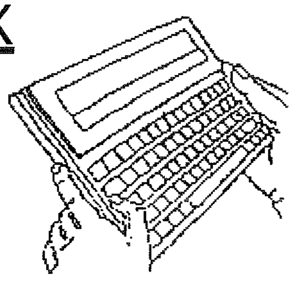
MOBILE: 07808 576118
E-MAIL: WNR@COMPUSERVE.COM

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL, PC AND MAC.

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS (ca. 20 HOURS) *



WE DO REPAIRS, PART EXCHANGES, AND BUY Z88's & PARTS

NEW! CAMBRIDGE Z88 WITH OZ4+512k Internal Ram £150

ALSO AVAILABLE, 512k RAMPACK £40

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	**£120**	**$180**
CAMBRIDGE Z88 COMPUTER	**£99**	**$130**
☆**RECONDITIONED Z88**	**£60**	**$90**
32k RAMPACK	£18	$25
128k RAMPACK	£38	$50
32k EPROM PACK	£18	$25
128k EPROM PACK	£24	$33
256k EPROM PACK	£55	$75
EPROM ERASER	£32	$43
PARALLEL PRINTER LEAD	£39	$50
SERIAL PRINTER LEAD	£12	$16
* **MAINS ADAPTOR (230vac; 6v,500ma)**	**£10**	
TOPPER (PROTECTIVE COVER)	£12	$16
CARRYING CASE (PLASTIC)	£10	$12

1/2 PRICE STARTER PACK
1 - Z88 ☆ PLUS
TRANSFER KIT
+ 32K RAM
+ 32K EPROM
£72 ONLY!!

FILE TRANSFER TO OTHER COMPUTERS

	UK	USA
SPECIAL Z88-QL SERIAL LEAD	£10	$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	$3
PCLINK KIT (For PCs)	£25	$35
Z88 TO MAC KIT	£25	$35
Z88 TO BBC KIT	£25	$35

} ALL WITH INSTRUCTION BOOKS, CABLES AND EPROMS AS REQUIRED!

POSTAGE UK UP TO £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

```
330   REMark '-' ->picture and memo fields
340   nvv%(y,0)=veld%(x+1,0):nvv%(y,1)=veld%(x+1,1)
350   newfield$(y)=name_of_field$(x+1)
360   y=y+1
370  END IF
380 END FOR x
390 CREATE_O #5;dbas_file$\nvv%
400 FOR x=0 TO noMemo-1:STNAME#5,x+1,newfield$(x):END FOR x
410 SAVE_NAMES #5
420 CREATE_DBAS_FILE
430 CLOSE_DATA#5
440 :
450 DEFine PROCedure OPEN_DBF_FILE
460 REPeat loop
470 foxprofile$=FILE_SELECT$(' Which DBF file? ',,,'dbf',25+800,,,2,2)
480 IF  '.dbf' INSTR foxprofile$ OR '_dbf' INSTR foxprofile$
490  EXIT loop
500  ELSE
510  AT#2, 4,2:PRINT#2, 'Extension should be .DBF! Press ENTER key to quit':INPUT#2,ab$ :
520  STOP
530 END IF
540 END REPeat
550 WHEN ERRor
560 IF ERNUM:CLOSE#3:REPORT_ERROR ERNUM,,,2:PRINT ERNUM,ERLIN:PAUSE:QUIT:END IF
570 END WHEN
580 IF FOP_IN(#3,foxprofile$)=0
590 ELSE
600  AT#2, 4,2:PRINT#2, "Can't open "&foxprofile$&" Press ENTER key to quit":INPUT#2,ab$ :STOP:
610 END IF
620 AT#2,4,2:PRINT#2,'Looking for details':
630 BGET#3, ID_dbf_file$(1 TO 1)
640 BGET#3, DATE_update$(1 TO 3)
650 BGET#3, TOT_rec$(1 TO 4)
660 BGET#3, POS_1field$(1 TO 2)
670 BGET#3, LEN_dbf$(1 TO 2)
680 BGET#3, RES_1$(1 TO 20)
690 IF  ID_dbf_file$(1 TO 1) INSTR CHR$(3)&CHR$(131)&CHR$(139)&CHR$(245)=0
700  CLOSE_AND_QUIT 'not a valid DBF file'
710 END IF
720  lr4=CODE(TOT_rec$(4 TO 4)):lr3= CODE(TOT_rec$(3 TO 3)):
730  lr2= CODE(TOT_rec$(2 TO 2)):lr1= CODE(TOT_rec$(1 TO 1))
740  lr4$=HEX$(lr4,8):lr3$=HEX$(lr3,8):lr2$=HEX$(lr2,8):lr1$=HEX$(lr1,8)
750  number_of_rec=HEX(lr4$&lr3$&lr2$&lr1$)
760  IF  numb_of_rec>32767:CLOSE_AND_QUIT 'Too many records for DBAS':END IF
770  lfich1=CODE(LEN_dbf$(2)):lfich2=CODE(LEN_dbf$(1))
780  lfich1$=HEX$(lfich1,8):lfich2$=HEX$(lfich2,8):lfich=HEX(lfich1$&lfich2$)
790  IF  lfich>32767:CLOSE_AND_QUIT 'Record too long for DBAS':END IF
800  lf=0:long=0:numb_of_fields=1
810  REPeat loop
820  IF EOF(#3):EXIT loop:END IF
830   BGET#3,REC_detail$(1 TO 32)
840   IF CHR$(13) INSTR REC_detail$(1 TO 1):lf=lf+1:END IF
850   sous=CODE(REC_detail$(17 TO 17)):long=long+sous
860   numb_of_fields=numb_of_fields+1
870   IF lf=1
880    ba$=INKEY$(#3):PTR_file=FPOS(#3)-32
890    long=long-sous:numb_of_fields=numb_of_fields-1
900    EXIT loop
910   END IF
920  END REPeat loop
930 END DEFine OPEN_DBF_FILE
940 :
950 DEFine PROCedure NAME_THE_FIELDS
960 DIM veld%(numb_of_fields,1),len_of_field%(numb_of_fields)
970 DIM name_of_field$(numb_of_fields,11),type_of_field$(numb_of_fields,1),len_field$(numb_of_fields,2)
980 BGET#3\32
990 noMemo=0:REMark counter for unwanted field types (picture,memo)
1000 lf=0:rec_len=0: :code_field_type=0
1010 REMark lf LineFeed= next record
1020 FOR boucle=1 TO numb_of_fields-1
1030  BGET#3,REC_detail$(1 TO 32)
1040  IF CHR$(13) INSTR REC_detail$(1 TO 1):lf=lf+1:END IF
1050   name_of_field$(boucle)= REC_detail$(1 TO 11)
1060   type_of_field$(boucle)= REC_detail$(12 TO 12)
1070   len_of_field%(boucle)= CODE(REC_detail$(17 TO 17))
1080  code_field_type= CODE(type_of_field$(boucle))
```

```
1090  SELect ON code_field_type
1100  =67 :REMark string field
1110   noMemo=noMemo+1
1120   name_of_field$(boucle)=name_of_field$(boucle)
1130   :veld%(boucle,0)= 0:veld%(boucle,1)=len_of_field%(boucle)
1140  =68 :REMark date field
1150   noMemo=noMemo+1
1160   name_of_field$(boucle)=name_of_field$(boucle) :
1170   veld%(boucle,0)= 2: veld%(boucle,1)=len_of_field%(boucle)
1180  =77 :REMark memo field
1190   veld%(boucle,0)= -1: veld%(boucle,1)=len_of_field%(boucle)
1200   name_of_field$(boucle)=name_of_field$(boucle)&"-" :
1210  =80 :REMark picture field
1220   veld%(boucle,0)= -1: veld%(boucle,1)= len_of_field%(boucle)
1230   name_of_field$(boucle)=name_of_field$(boucle)&"-"
1240  =78,70 :REMark  numeric field
1250   noMemo=noMemo+1
1260   name_of_field$(boucle)=name_of_field$(boucle) :
1270   veld%(boucle,0)= 3: veld%(boucle,1)= len_of_field%(boucle)
1280  =76 :REMark  logical field
1290   noMemo=noMemo+1
1300   name_of_field$(boucle)=name_of_field$(boucle)&"$"
1310   veld%(boucle,0)= 0:veld%(boucle,1)=len_of_field%(boucle)
1320  =REMAINDER
1330   REMark should never get here
1340  END SELect
1350   REMark get rid of unwanted char in field name
1360   intnom$=name_of_field$(boucle):name_of_field$(boucle)=ALPHANUMONLY(intnom$)
1370   len_field$(boucle)= CODE(REC_detail$(17 TO 17))
1380   sous=CODE(REC_detail$(17 TO 17)):rec_len=rec_len+sous
1390   IF lf=1
1400    ba$=INKEY$(#3):PTR_file=FPOS(#3)-32
1410    rec_len=rec_len-sous+1
1420    EXIT boucle
1430   END IF
1440 END FOR boucle
1450 END DEFine NAME_THE_FIELDS
1460 :
1470 DEFine FuNction ALPHANUMONLY(a_n_str$)
1480 LOCal x,a$
1490 REMark get rid of unwanted char in field names (eg. trailing chr$(0))
1500 a$=""
1510 IF LEN(a_n_str$)=0:RETurn a_n_str$:END IF
1520 FOR x=1 TO LEN(a_n_str$)
1530  IF a_n_str$(x TO x) INSTR "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$@-"
1540   a$=a$&a_n_str$(x TO x)
1550  END IF
1560 END FOR x
1570 RETurn a$
1580 END DEFine ALPHANUMONLY
1590 :
1600 DEFine PROCedure CREATE_DBAS_FILE
1610 LOCal x,bblock,field_type,Field_len
1620 Str_ptr=1:x=1:dbs_fldn=1 :bblock=0
1630 DIM record$(rec_len),rec_st$(1):fld_cont$=''
1640 BGET#3\PTR_file
1650 WHEN ERRor
1660 erreur=ERNUM:BEEP 1000,30,30:AT#2,4,0: CLS#2,3:AT#2,24,2:PRINT#2, ERLIN,' File probably corrupted  Press
     a key to quit'
1670 INPUT#2,ab$:REPORT_ERROR erreur,,,2:AT 25,0:PRINT ERLIN,ERNUM:PAUSE:CLOSE#1,#2,#3,#5:STOP
1680 END WHEN
1690  BLOCK#2, 106,14,98,98,2:INK#2,2:AT#2, 10,18:PRINT#2,'Please wait'
1700  REPeat loop
1710  IF EOF(#3):BEEP 100,100:EXIT loop :END IF
1720    BGET#3,rec_st$(1 TO 1)
1730  IF EOF(#3):BEEP 100,100:EXIT loop :END IF
1740  BGET#3,record$(1 TO rec_len)
1750   APPEND#5
1760  REPeat loopin
1770  IF Str_ptr>=rec_len:EXIT loopin:END IF
1780   field_type=veld%(x,0):Field_len= veld%(x,1)
1790   fld_cont$=record$(Str_ptr TO Str_ptr+Field_len-1)
1800   SELect ON field_type
1810   =-1:REMark  to get rid of memo and picture fields
1820   =0 :REMark string field
1830    SET#5,dbs_fldn,fld_cont$:UPDATE#5:dbs_fldn=dbs_fldn+1
```

```
1840   =1 :REMark integer
1850    IF alltrim(fld_cont$)='':fld_cont$='0':END IF
1860    SET#5,dbs_fldn,fld_cont$ :UPDATE#5:dbs_fldn=dbs_fldn+1:
1870   =2 :REMark  date field/long integer
1880    REMark if date field empty
1890    IF alltrim(fld_cont$)=""
1900     fld_cont$='19610101'
1910    END IF
1920    yr=fld_cont$(1 TO 4):mt=fld_cont$(5 TO 6):dy=fld_cont$(7 TO 8)
1930    SET#5,dbs_fldn,DATE(yr,mt,dy,0,0,0) :UPDATE#5:dbs_fldn=dbs_fldn+1:
1940   =3 :REMark numeric
1950    IF alltrim(fld_cont$)='':fld_cont$='0':END IF
1960    SET#5,dbs_fldn,fld_cont$:UPDATE#5:dbs_fldn=dbs_fldn+1:
1970   =REMAINDER
1980    REMark should never get here
1990   END SELect
2000  Str_ptr=Str_ptr+Field_len:x=x+1
2010  END REPeat loopin
2020   bblock=bblock+1: BLOCK  bblock,10,100,100,2
2030   IF bblock>100:BLOCK bblock,10,100,100,0:bblock=1::AT#2, 10,18:PRINT#2,'Please wait':END IF
2040  Str_ptr=1:x=1 :dbs_fldn=1
2050 END REPeat loop
2060 BLOCK#2, 100,10,100,100,7:INK#2,2:AT#2,10,18:PRINT#2,'Press a key':PAUSE
2070 END DEFine CREATE_DBAS_FILE
2080 :
2090 DEFine FuNction alltrim(xxx$)
2100 LOCal a$,b$,l,x
2110 REMark get rid of chr$(32)
2120 a$=" ":b$="":l=LEN(xxx$)
2130 IF xxx$="":RETurn "":END IF
2140 FOR x =1 TO l
2150  IF xxx$(x)==a$
2160  ELSE
2170   b$=b$&xxx$(x)
2180  END IF
2190 END FOR x
2200 RETurn b$
2210 END DEFine alltrim
2220 :
2230 DEFine PROCedure CLOSE_AND_QUIT (mesg$)
2240 LOCal ok$,x
2250 ok$=ITEM_SELECT('Message Box',mesg$,'Press a key',,,,,2)
2260 FOR x=5 TO 1 STEP -1:CLOSE#x:END FOR x
2270 STOP
2280 END DEFine CLOSE_AND_QUIT
2290 :
2300 DEFine PROCedure OPEN_DBAS_FILE
2310 REPeat loop
2320 dbas_file$=FILE_SELECT$(' To which DBAS file? ','test2','ram2_','_dbs',25+800,,,2,2)
2330 IF  '_dbs' INSTR dbas_file$  OR '.dbs' INSTR dbas_file$
2340  EXIT loop
2350  ELSE
2360  AT#2, 4,2:PRINT#2, 'Extension should be _dbs or .dbs! Press ENTER key to quit':INPUT#2,ab$ :
2370  CLOSE#3 :REMark close opened dbf file
2380  STOP
2390 END IF
2400 END REPeat loop
2410 END DEFine OPEN_DBAS_FILE
```

Enhancements revisited

Marcel Kilgus

In issue 6/volume 6 I described some new features I was playing around with, like the fast memory and of course the enhanced pointer environment. In the meantime you can check out both goodies in the latest SMSQ/E for QPC version (2e99).

In the following paragraphs I'd like to have a closer look at some aspects and problems. You will notice that this article is at times fairly technical but there are also some things mentioned that the "common user" might be interested in, therefore it might be a good idea to ignore the incomprehensive passages and just read on.

Fast memory

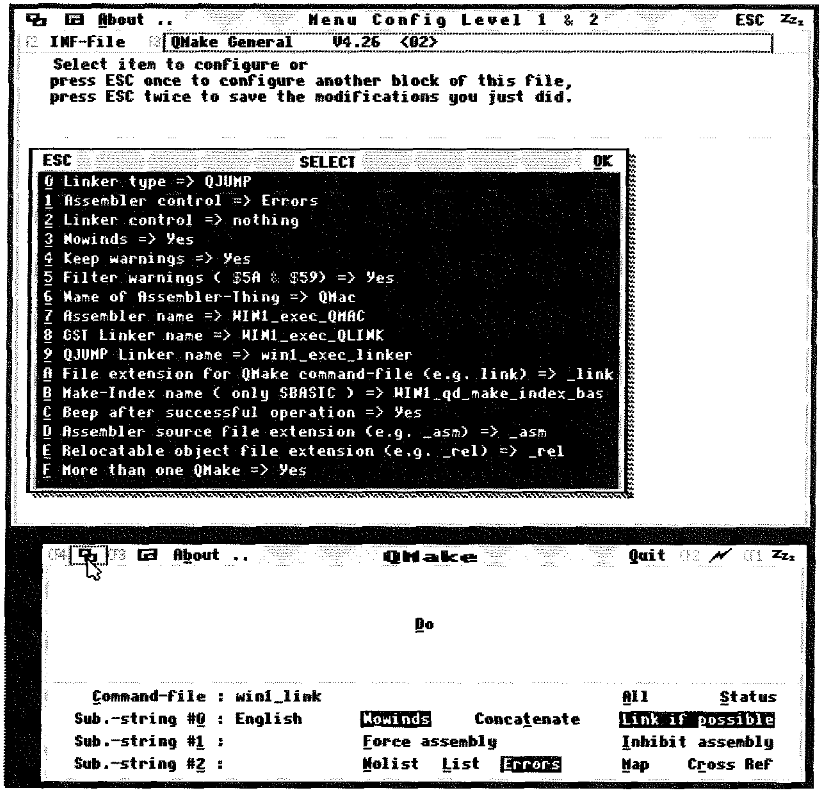
Fast memory is working fairly well, the general speed improvement of disc accesses is very noticeable and was certainly worth the amount of work that went into it. I am currently employing 1MB of "slow" memory

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://smsq.j-m-s.com

QMAKE

QMAKE, a pointer-driven MAKE program which works on LINK files suitable for QLINK and Tony Tebby's own linker, will not only allow you to link easily language-specific or machine-dependant versions, it also takes care of CCT files etc.
The snapshots show some of the configuration features and the QMAKE window when opened.



```
About ..          Menu Config Level 1 & 2          ESC
INF-File   QMake General    V4.26  <02>
 Select item to configure or
press ESC once to configure another block of this file,
press ESC twice to save the modifications you just did.

ESC                    SELECT                        OK
0 Linker type => QJUMP
1 Assembler control => Errors
2 Linker control => nothing
3 Nowinds => Yes
4 Keep warnings => Yes
5 Filter warnings ( $5A & $59) => Yes
6 Name of Assembler-Thing => QMac
7 Assembler name => WIN1_exec_QMAC
8 GST Linker name => WIN1_exec_QLINK
9 QJUMP Linker name => win1_exec_linker
A File extension for QMake command-file (e.g. link) => _link
B Make-Index name ( only SBASIC ) => WIN1_qd_make_index_bas
C Beep after successful operation => Yes
D Assembler source file extension (e.g. _asm) => _asm
E Relocatable object file extension (e.g. _rel) => _rel
F More than one QMake => Yes
```

```
About ..              QMake                    Quit

                         Do

  Command-file : win1_link                          All        Status
Sub.-string #0 : English     Nowinds   Concatenate  Link if possible
Sub.-string #1 :             Force assembly         Inhibit assembly
Sub.-string #2 :             Nolist  List  Errors   Map     Cross Ref
```

QMAKE can contract to a button whilst working. Ideal for linking SMSQ/E ...

QMAKE Special SMSQ/E Summer Offer
(valid until Aug.30, 2002) **EUR 19,90**

TERMS OF PAYMENT

Postage and package **[Germany]** EUR 4,86 (if total value of goods is up to EUR 25,- then only EUR 3,32).
[Europe] £4,50 / EUR 7,41 (if total value of goods is under £15 / EUR 25,56 then only £3 / EUR 4,86).
[Overseas] between US$7.50 (1 item) and US$17.50 (maximum).
Cheques in EURO and International Money Order in EURO (both have to be drawn on German banks).
We accept Internation Reply Coupons as payment (can bo bought at any post office): 1 IRC = EUR 1.02 redeeming value.
Exact prices are given in EUR ... the price in £ and/or US$ are approx. conversion based on the exchange rate at the time of creation of this ad. Prices may be adjusted in case the exchange rate falls considerably out of this range in both directions!
We can charge your credit cards in £'s, US$, EURO or in DM - please state the currency you prefer.
Please do not send any UK bank cheques in £ - our banks have increased the fee for handling them by 600% (no joke!) so we cannot accept them anymore, unless you add £6 for clearing the cheque.
E&OE.

EUROCARD. VISA

for slaving etc. and the rest as "fast" memory. This of course (and unfortunately) has some impact on existing code. The most obvious problem is the one of determining the amount of free memory. A lot of applications use the formula

sys_sbab - sys_fsbb - $400

to determine this, which essentially calculates the size of the gap between the common heap and the SuperBASIC memory area (minus the minimum space required for slaving). This method now fails and wrong and much too low values are returned. And it has some other drawbacks anyway (like failing completely when the memory is fragmented) and therefore I encourage application developers to use the system trap sms.frtp (MT.FREE in QDOS slang) instead. On "traditional" SMSQ/E systems it essentially does the same calculation as outlined above, on the fast memory enhanced system it correctly returns the size of the biggest free block available. CueShell as a commercial application used it all along and therefore didn't need any change to show the correct value under the new QPC.

As, if I remember correctly, Cueshell also works under QDOS I expect the call to work fine there, too. To be sure I had a look at the QDOS classic sources (which essentially equals the JS rom in the core) and apart from one little bug (one register. namely D2. doesn't seem to get initialised anywhere) it looks ok. The worst case scenario for the bug is that only the calculation mentioned above is carried out instead of finding the largest block in a fragmented memory scenario, so no big deal. As a workaround one could also clear D2 before calling the trap. Again, the bottom line is to use sms.frtp to determine the amount of free memory. SBASIC programmers don't have any problem because I have already modified the FREE_MEM function to use sms.frtp.

On my system I use a little freeware utility to display the current free memory in the buttom frame. It's written by Oliver Fink and called "Free memory display" or FreeMemBt. I have patched the program to use sms.frtp and by the way also increased the window size (it wasn't able to show more than 9999KB of ram). If anybody wants the patched version, please contact me.

Another problem is that Sysmon, which is probably in use by more people, is not prepared for the new situation and quits with a nasty beep. The reason is that it doesn't like the idea of a completely empty common heap and therefore thinks the heap is corrupt. I have aquired the sources to Sysmon and a new release will be made soon (by a nice coincidence Sysmon has another problem which, when fixed, cancels out the first one because then the common heap isn't empty anymore while Sysmon runs, i.e. one won't lose out on protection with the new version). The very same Sysmon code is also incorporated into the QPAC2 "jobs" window, where, with fast memory enabled, only "corrupt memory" is shown. Together with the new Sysmon I will also prepare a new QPAC2 version, so just make sure to update once those two new versions get released.

Now to the more serious stuff. I got a report that compiling large programs using QD/QBASIC doesn't work anymore and that there might be some problem with Qliberator.

Furthermore there is another report that Abacus doesn't load anything anymore when QPC is set to use more than 15MB of RAM and that PWFile can completely crash the machine when trying to do anything with it. The issues are currently under investigation and I expect that they will finally be sorted out. If you encounter any strange complications because of the introduction of fast memory please contact me. Sometimes it takes some time for me to answer but all feedback is highly appreciated, believe me.

The last question I got was whether the new memory scheme will also be available on other platforms. The answer to this is yes, I think so. I will make the necessary code available and as it is hardware independent it should not be too big a problem to incorporate into other SMSQ/E versions. At least I expect the Qx0 to highly profit from it. Whether it makes much sense on (Super)GoldCards I'm not really sure, but anyway, the code will be there so it can be done if anybody bothers to do it.

High colour pointer environment

The new QPC2 includes the new WMAN version 1.54. It is identical to 1.53 apart from the fact that it includes the first stage of the high colour addition, i.e. it can now interpret the new colour format which by the way also got one enhancement:

%00000000cccccccc	exactly as before
%00000001pppppppp	palette
%00000010pppppppp	system palette (not implemented yet)
%00000011gggggggg	grey scale
%01ssxxxxxyyyyyy	palette stipple. see below ***(New!)***
%1rrrrrgggggbbbbb	15 bit RGB

Stipple format (as proposed by George Gwilt):
s = stipple code (0 = dot, 1 = horizontal, 2 = vertical, 3 = checkers)
x = stipple colour y = main colour

As you can see it now includes a new stipple format. Initially I was against introducing any stipples but in the end I acknowledged that sometimes a little grainy colour can look good. Due to the space restriction only the first 64 entries of the palette can be used for stipples, but fortunately those contain a complete subset of the palette, i.e. the other 192 colours only fill the gaps between the ones defined in the first 64 entries. The new format can be used in every location where a colour is needed within the WMAN structures, the only exception being the colour of the scroll/ pan arrows. Due to their nature those need a completely different treatment and therefore must be implemented separately. This will be done in a later stage.

System palette

As a remainder, the "system palette" is a palette I thought up that holds the default colours for different window items, like the title bar colour, the button colour, the window background colour etc. You might have noticed that I wrote "not implemented yet" earlier when it came to the system palette, this is because it's still on my drawing-board. And that's what I have thought up so far:

The format of the palette is the colour definition mentioned earlier, i.e. it consists of an array of 16bit values. The only difference I will probably make is that a system palette entry can not point to another system palette entry to prevent a possible endless loop. Or I might just limit the recursion to a certain level, like 3.

Of course there must be an interface to read and change the palette entries. For this two new WMAN vectors named `wm.setsp` and `wm.getsp` will be introduced. You can see a detailed draft of the API functions in the separate box.

So those functions can be used to read and set the entries while the system is running. Of course already open windows won't change their colour this way but windows opened afterwards will already employ the new definitions. Remains the question what values the system uses after the boot process. There I have been thinking of introducing a standard Config Level 2 block that contains all palette entries. It won't make much sense to change the configuration using MenuConfig for example (at least not until it gets extended with a colour picker), but custom made configuration programs could parse the widely known structure and patch in new values at the appropriate places. Of course the block will already be shipped with a default colour scheme that could just be overwritten on boot time through the mentioned vector.

Speaking of custom made configuration programs, that's the only part of this lot I won't be able to provide. It's clear that such a program will be necessary, so I'm relying on you all out there to write the 3rd party tools to do it. I just might be able to provide SBASIC wrappers for the wm.setsp and wm.getsp vectors.

Last but really not least, what items are there to include in the palette? So far I was able to come up with the following ones, any additional ideas are very welcome!

Window background
Window foreground
Window border
Subwindow background
Subwindow foreground
Subwindow border
Title bar background
Title bar foreground
Button available background
Button available foreground
Button selected background
Button selected foreground
Button unavailable background
Button unavailable foreground
Button border
3D object background
3D object foreground
Pan/scroll bar background
Pan/scroll bar section
Pan/scroll bar arrow
Tooltip background
Tooltip foreground
Tooltip border

"Foreground" essentially means "ink colour". So that's 23 items total so far. I hope that I have covered most parts, but in the end YOU are the ones that I hope will pick up this idea and use it in your programs. So if anything's missing, tell me!

As said all the system palette stuff described above did not result in any code so far (well, actually it did, but that's just to help me thinking), i.e. it can all still be changed. By defining a new colour format and API calls I'm getting a bit into Tony's role. Of course in contrast to Tony I don't really have the power to decide this on my own, therefore this "public announcement". I'm open for any discussion regarding my proposals.

Aurora drivers

The feedback on high colour Aurora drivers was at first very disappointing, within the first weeks I got only 2 mails that stated interest. Then however I got one mail from a fellow German QL user that did make the task appear a little bit more worthwhile. This means that I'm generally inclined to do it, but as I'm pretty short of time again (work and important exams coming up again) I can't say anything definite yet. Some more feedback on this issue might help though (hint, hint).

[Actually, Marcel, there were comments it was only in QL Today and only in your article - so people may not have noticed it - we shall put it into the News section and see if that gives more feedback - Jochen]

Finally

On Jochen's last birthday I sent him a coupon as a present that granted him one QL Today article from me when he's short of material. And well, here it is, as always finished much too late but on the other hand there wasn't much time to start with anyway. At least I'm glad that you made it till here (unless you're one of those unusual people that start reading books and articles at the end). I hope there was something in it for everybody and if it's only the answer to "why doesn't Sysmon work anymore on the new QPC". In case you don't know it already, my e-mail address is **marcel@kilgus.net**, you can send all your comments and the pity for my bad writing style there (to my excuse, it's currently 4am). Anyway, have a nice day.

```
Vector $7C                                                    WM.SETSP

     Set system palette entries

Call parameters                      Return parameters

D1.w start index                     D1    preserved
D2.w number of elements              D2    preserved
                                     D3+   all preserved

A0                                   A0    preserved
A1   pointer to palette entries      A1    preserved
A2                                   A2    preserved
A3                                   A3    preserved
A4                                   A4    preserved
A5   not used by any routine
A6   not used by any routine

Error returns:
     IPAR    Illegal index number / invalid number of elements
```

Set the entries of the system palette to the values in the buffer, beginning with the index in D1 (counting from 0) and ending with the index D1 + D2 - 1.

The buffer must hold an array of words with the colour values of the different items. The colour format is the standard WMAN colour format as described elsewhere.

```
Vector $80                                                    WM.GETSP

     Read system palette entries

Call parameters                      Return parameters

D1.w start index / -1                D1.w preserved / item count
D2.w number of elements              D2   preserved
                                     D3+  all preserved

A0                                   A0   preserved
A1   pointer to entry buffer         A1   preserved
A2                                   A2   preserved
A3                                   A3   preserved
A4                                   A4   preserved
A5   not used by any routine
A6   not used by any routine

Error returns:
     IPAR    Illegal index number / invalid number of elements
```

Copies entries of the system palette into the given buffer, beginning with the index in D1 (counting from 0) and ending with the index D1 + D2 - 1. The buffer must be big enough to hold all requested entries.

If D1 is given as -1 the function just returns the number of items held in the system palette. This can increase when more items get defined in new WMAN versions. This is guaranteed to be below 256.

What Machine am I running on?

Dilwyn Jones

Back in issue 5 of volume 6 of QL Today, we printed an article on extracting data from the system to identify computer details such as processor type, display type, computer type and so on. While useful, this article was written from an SMSQ/E and SBASIC standpoint and several people have asked if similar code can be achieved from QDOS.

While my initial appeal for information drew almost complete silence in return so I'll presume either people don't know how, or are not interested.

A few facts did come to light though, and these are printed below. The Hermes manual documents how to test for the presence of a Hermes chip (the TF Services replacement chip for the IPC8049 second processor in the QL) and the Aurora Technical Documentation shows how to test for an Aurora board.

The following method of testing for an Aurora board is quoted from the Aurora Technical manual:

The Aurora can be detected at reset as follows:
At reset the Aurora mimics the 8301 ULA and will automatically set itself into 512x256 mode 4, SCR0 active. Because of the hardware remapping of SCR0 into the high-res screen area, anything written into the first 128 bytes of SCR0 can be read in the first 128 bytes of the high-res area, BUT NOT THE OTHER WAY AROUND! because the GC/SGC shadows only SCR0 and SCR1 and not the high-res area. Write only to SCR0 and read only from the high-res area to test for Aurora, not the other way around. In the previous specifications this step was replaced by a test for RAM at $4C0000, I do not recommend that because the address will change with our next product, and the current address will most likely hold ordinary RAM and not Aurora screen RAM. After the presence of a high-res area is detected, the amount of the high-res area RAM should be tested. This will be either 240 kbytes or 128 kbytes. If the amount is 240 kbytes, Aurora is detected. If the amount is 128 kbytes, a LCD board is detected.

The following method of testing for the presence of a Hermes chip in assembly language on a QL is quoted from the Hermes manual:

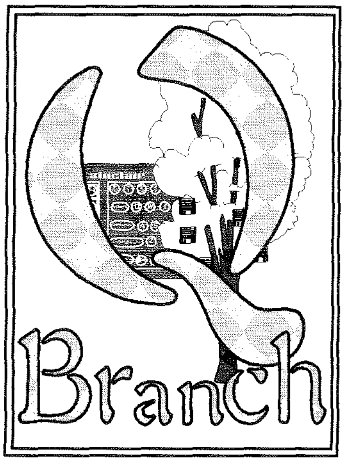
```
;data blocks for MT.IPCOM (or SMS.HDOP) trap #1, d0=$11
hermon   dc.b      $0C       ;MDRS_CMD
         dc.b      $01       ;1 parameter
         dc.l      $00000000 ;bit mask %00=send 4 bits
         dc.b      $09       ;parameter - to turn Hermes on
         dc.b      $01       ;bit mask %01
;no return value

hermis   dc.b      $0F       ;TEST_CMD
         dc.b      $01       ;1 parameter
         dc.l      $00000002 ;bit mask %10=send 8 bits
         dc.b      $A3       ;value sent
         dc.b      $02       ;bit mask %10=send 8 bits
;return value in d1.b: $A3= not Hermes, $5C= Hermes is there

hermtest lea       hermon,a3 ;code to turn on Hermes
         moveq     #$11,d0   ;MT.IPCOM (SMS.HDOP)
         trap      #1
         lea       hermis,a3 ;TEST_CMD
         moveq     #$11,d0   ;MT.IPCOM (SMS.HDOP)
         trap      #1
         cmpi.b    #$5C,d1   ;is Hermes there?
         bne.s     noherm    ;Hermes not there
yesherm                      ;Hermes is there - appropriate
code here
.........
noherm                       ;code to handle no Hermes here
.........
```

SMSQ/E must have internal routines to test for the presence of the various processors and hardware types, and now that the source files for SMSQ/E is being made available to developers, it is likely that a knowledgeable machine code programmer could extract the necessary code and perhaps come up with either an assembler listing or set of extensions for QDOS to test for given hardware types. Any offers, anyone? Of course, given that SMSQ/E is modular and different versions of the operating system exist for different machines, certain details may well be hard coded into the version concerned rather than tested for as such, in which case there would be a problem.

"What Hardware" Listing *from Jack Mitchell*

```
10000 REMark *****************************
10010 REMark Version: V1.0
10020 REMark Date: 20/Sep/1997
10030 REMark Author: J.D.Mitchell
10040 REMark EMAIL: Jack@home9999.demon.co.uk
10050 REMark Change: 1) Default system vars base is at 160K not 128K
10060 REMark 2) Add change control header
10070 REMark *****************************
10080 REMark Version: 1.1
10090 REMark Date: 20/Sep/1997
10100 REMark Author: J.D.Mitchell
10110 REMark EMAIL: Jack@home9999.demon.co.uk
10120 REMark Change: 1) Use peek(!!addr) etc. when possible
10130 REMark *****************************
10140 REMark Version: 1.2
10150 REMark Date: 28/Nov/1997
10160 REMark Author: J.D.Mitchell
10170 REMark EMAIL: Jack@home9999.demon.co.uk
10180 REMark Change: 1) Corect error in change history
10190 REMark Change: 2) Report QDOS versions as ROM names not versions
10200 REMark *****************************
10210 DEFine FuNction DISPLAY_WIDTH
10220   IF os$='SMSQE' THEN RETurn SCR_XLIM
10230   RETurn scr__lim%('scr_','x256a0x0')&&-2
10240 END DEFine
10245 :
```

Q Branch



Text 87

£ 79.00

Typset 94	£ 29.00
Fountext 94	£ 39.00
2488 drivers	£ 29.00
Epson ESC/P2 drivers	£ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handliing the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

PROGRAMMING

QD 98	£ 45.00
QD + QBasic	£ 59.00
QD + Qliberator + QBasic	£ 100.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 30.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 15.00
QMon / JMon	£ 22.00
Basic Linker	£ 19.00
DISA 3	£ 31.00
QMenu	£ 14.00

Summer Sizzlers

Q Branch have managed to get a few more boxes of IBM ED disks. This is the last of these from the current source and they are now becoming very rare.

If you are using ED disks stock up now. They are available in Plastic Boxes of 10 for £ 7.50. There are not very many so call us now if you want some.

We also have a complete QL system, consisting of Aurora, superHermes, Qubide, Super Gold Card etc in a tower case as well as a couple of Auroras and Qubides. Call us for details.

Qubide IDE intefaces : £ 45.00

Aurora Motherboards : £ 60.00

New ED Disks

£ 7.50 per box

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
(Includes Pointer and non-pointer driven versions)
(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)
Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC

Convert text files from QL to PC formats and much more !

Only £ 10.00 Now with HTML support !

QL Rhymes £ 10.00

Rhyming Dictionary

Spelling Crib : PD program £ 1.50 +SAE
or Free if you buy all three programs

UTILITIES

FiFi 2	£ 18.00
QSup	£ 28.00
QSpread v 3	£ 48.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 28.00

We are currently out of stock of the SuperBasic Reference Manual Place your order now to get one as soon as it is reprinted

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !
£ 40.00
+ postage

News on the SuperBasic Reference Manual Coming soon !

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

Tel +44 (0) 1273-386030 fax +44 (0) 1273-430501
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

New Fax Number

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor

Demo version £ 1.50 + postage
Full Registered version £ 18.00

Version 2.03 available now !

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

QXL II	**£ 100.00**
Recycled superHermes	£ 65.00 *
Recycled Gold Card	£ 45.00 *
Recycled Aurora	See over*
Qubide	See over
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

Qubide upgrades to version 2.01 £ 8.00

SMSQ/E

Gold Card / Atari / QXL Version

Now Only £ 32.00

Various Atari versions : call for details

QPC 2 v3 full colour version!
Now Only £ 65.00

Upgrades from QPC 1 £ 46.00 (return master disk)
Upgrades from QPC 2 v1 £ 30.00
Upgrades from QPC 2 v2 £ 14.00
If Bought/upgraded after 15/10/01 £ 5.00

Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E or QPC 2 upgrade !

Q Branch Programs

The Knight Safe 3	£ 35.00	Q - Route v1.08C Route finding programme	£ 25.00
upgrades from previous versions	£ 5.00	The Fractal Collection	£ 35.00
Q - Count Pointer driven home accounting	£ 25.00		

Post and Packing is included with Software Prices

We can accept payment by Visa, Mastercard and Switch. We also accept cheques drawn on a UK Bank in Sterling.
You can also pay directly into our bank in Euros.
Barclays Bank Acc# 79016888 sort code :20-71-02 Name :QBranch)

Checking for Minerva is easy: just test IF VER$='JSL1'. Similarly, SMSQ or SMSQ/E (well, SBASIC actually) can be tested for with IF VER$='HBA'. The listing below includes a method of distinguishing between SMSQ (QXL) and SMSQ/E.

Pointer environment can be tested for by calling the IOP.PINF trap (trap #3 with d0=$70, d3.w=timeout value and a0=window channel ID. If there is no pointer interface, d0 will contain the invalid parameter error code, while if d0=0 (no error) pointer environment is present with the version in d1.l - further information on page 57 of Jochen Merz's QPTR guide.

"What Hardware" Listing

A nice kind person also sent me a listing in SuperBASIC by J. D. Mitchell which sets out to provide similar information on all systems by providing a set of prcoedures and functions to report on machine type and so on.

The listing itself is fairly complex and I'll try to explain most of what it does.

The procedure SETUP_TEST at line 11770 ensures that Toolkit 2 is active, then calls SETUP_READ to test for information present in the system indicating details of the hardware present. Finally, SETUP_SHOW is called to list the details to the screen.

To some extent, this listing relies on the presence of two little documented system variables at offsets 161 (sys_ptyp, at hex A1) and 167 (sys_mtyp, at hex A7) within the system variables area. These were documented on page 28 of

```
10250 DEFine FuNction display_height
10260   IF os$='SMSQE' THEN RETurn SCR_YLIM
10270   RETurn scr__lim%('scr_512x','a0x0')
10280 END DEFine
10285 :
10290 DEFine FuNction scr__lim%(prefix$,sufix$)
10300   LOCal h_max%,h_min%,h%,loop
10310   h_max%=1280
10320   h_min%=256
10330   REPeat loop
10340     h%=(h_max%+h_min%+1)DIV 2
10350     IF h%>h_max% THEN h%=h_max%
10360     IF FTEST(prefix$&h%&sufix$)=0 THEN
10370       h_min%=h%
10380     ELSE
10390       h_max%=h%-1
10400     END IF
10410     IF h_min%=h_max% THEN RETurn h_max%
10420   END REPeat loop
10430 END DEFine
10435 :
10440 REMark mimics peek_w(!!addr)
10450 DEFine FuNction peek_w_sys(addr)
10460   IF os$='QDOS' THEN
10470     RETurn PEEK_W(addr+sys_vars_base)
10480   END IF
10490   RETurn PEEK_W(!!addr)
10500 END DEFine
10505 :
10510 REMark mimics peek(!!addr)
10520 DEFine FuNction peek_sys(addr)
10530   IF os$='QDOS' THEN
10540     RETurn PEEK(addr+sys_vars_base)
10550   END IF
10560   RETurn PEEK(!!addr)
10570 END DEFine
10575 :
10580 REMark mimics peek_l(!!addr)
10590 DEFine FuNction peek_l_sys(addr)
10600   IF os$='QDOS' THEN
10610     RETurn PEEK_L(addr+sys_vars_base)
10620   END IF
10630   RETurn PEEK_L(!!addr)
10640 END DEFine
10645 :
10650 DEFine PROCedure setup_read
10660   LOCal hardware_code,cpu_code,display_code,sys_vars_base,
        monitor_code,display_register
10670   IF VER$<>'HBA' THEN
10680     IF VER$='JSL1' THEN
10690       os$='Minerva'
10700       version$=VER$(1)
10710       sys_vars_base=VER$(-2)
10720       lang$='GB'
10730     ELSE
10740       os$='QDOS'
10750       version$=VER$
10760       sys_vars_base=(128+32)*1024
10770       IF LEN(version$)<=2 THEN
10780         lang$='GB'
10790       ELSE
10800         lang$=version$(3 TO)
10810       END IF
10820     END IF
10830   ELSE
10840     IF FTEST('history_1024')<>0 THEN
10850       os$='SMSQ'
10860     ELSE
10870       os$='SMSQE'
10880     END IF
```

```
10890     sys_vars_base=VER$(-2)
10900     version$=VER$(1)
10910     lang$=LANGUAGE$
10920    END IF
10930    REMark mimics  PROCESSOR   provided by SMSQE
10940    cpu_code=(peek_sys(161) DIV 16)*10
10950    REMark mimics  MACHINE     provided by SMSQE
10960    hardware_code=peek_sys(167)&&31
10970    REMark mimics  DISP_TYPE   provided by SMSQE but gives 6
         for VGA on a QXL
10980    display_code=peek_sys(167) DIV 32
10990    display_register=peek_sys(52)
11000    freq=peek_w_sys(168)
11010    SELect freq
11020      =50,60
11030      =REMAINDER :freq=50
11040    END SELect
11050    ram_top=peek_l_sys(32)
11060    monitor_code=peek_sys(50)
11070    REMark Convert to text
11080    SELect ON hardware_code
11090      =0:hardware$='Standard QL'
11100      =1:hardware$='Atari'
11110      =2,3:hardware$='Mega/RTC ST'
11120      =4,5:hardware$='Stacy'
11130      =6,7:hardware$='STE'
11140      =8,9:hardware$='Mega STE'
11150      =10,11:hardware$='GOLD card'
11160      =12,13:hardware$='Super Gold card'
11170      =16:hardware$='Falcon'
11175      =17:hardware$='Q40 or Q60'
11180      =24:hardware$='TT 030'
11190      =28:hardware$='QXL'
11200      =30:hardware$='QPC'
11210      =31:hardware$='QLAY'
11220      =REMAINDER :hardware$='Unknown HW('&hardware_code&')'
11230    END SELect
11240    SELect hardware_code
11250      =1,3,5,7,9:add_on$='Blitter'
11260      =11,13:add_on$='Hermes'
11270      =REMAINDER :add_on$=''
11280    END SELect
11290    SELect ON monitor_code
11300      =0:monitor$='Monitor'
11310      =1:monitor$='625 line TV'
11320      =2:monitor$='525 line TV'
11330      =REMAINDER :monitor$='Unknown Monitor('&monitor_code&')'
11340    END SELect
11350    display_mode=4+(display_register&&8)DIV 2
11360    display_page=display_register DIV 128
11370    SELect display_code
11380      =0:display$='Standard'
11390      =1:display$='Monochrome'
11400      =2:display$='Extended 4 emulator'
11410      =4:display$='QMVE'
11420      =5:display$='Aurora'
11430      =6:display$='VGA'
11440      =REMAINDER :display$='Unknown display('&display_code&')'
11450    END SELect
11460    SELect cpu_code
11470      =0:cpu$='68000/68008'
11480      =10:cpu$='68010'
11490      =20:cpu$='68020'
11500      =30:cpu$='68030'
11510      =40:cpu$='68040'
11515      =60:cpu$='68060':REMark added by DJ
11520      =REMAINDER :cpu$='Unknown CPU('&cpu_code&')'
11530    END SELect
11540 END DEFine
11545 :
```

Volume 6 Issue 5 of QL Today. There is one additional value possible for sys_mtyp - after $1e or decimal 30 for QPC, we should add $1F or decimal 31 for the QLay emulator.

These systems variables are not supported on all versions of QDOS, which can make life a bit harder for us.

QL systems with Gold Card or later expansions patch QDOS to support this facility. Earlier QL systems and some emulators which use QDOS do not support this facility and will not unless someone comes up with either a method of patching QDOS ROM images or provides some code to add the facility to QDOS, as suggested above.

10210 to 10240: Display_Width is a procedure which tests for the maximum possible display width on this system. If it detects that the operating system is SMSQ/E (test for SMSQ or SMSQ/E is VER$='HBA' and distinguishes between SMSQ and SMSQ/E by the FTEST('history_1024') in line 10840), it uses SBASIC's SCR_XLIM function to return the maximum possible window height on the current display. If not SMSQ/E, the function scr__lim% is called to try to establish how large a SCR channel may be opened using FTEST. Please note that if using Norman Dunbar's DJToolkit, you may have to rename the Display_Width function defined in line 10210 as that toolkit includes an extension with a similar name, but performing a different function.

10250 to 10280: Display_Height is a similar routine for establishing the maximum possible window height for the current display.

```
11550 DEFine PROCedure setup_show(chan%)
11560   PRINT#chan%,'Running ';os$;
11570   IF os$=='QDOS' THEN
11580     PRINT#chan%,' ';version$;' ROM'
11590   ELSE
11600     PRINT#chan%,' V';version$
11610   END IF
11620   PRINT#chan%,'On a ';cpu$;' powered ';hardware$
11630   PRINT#chan%,'With ';INT(ram_top/1024)-128;'KByte RAM';
11640   IF add_on$<>'' THEN PRINT#chan%,' and ';add_on$
11650   PRINT#chan%,\'Using ';display$;'
';DISPLAY_WIDTH;'x';display_height;' on a ';monitor$
11660   PRINT#chan%,'Displaying ';position_name$(display_page+1);'
page in mode ';display_mode
11670   PRINT#chan%,'Country ';lang$;' @';freq;'Hz'
11680 END DEFine
11685 :
11690 DEFine FuNction position_name$(n)
11700   SELect n
11710     =1:RETurn '1st'
11720     =2:RETurn '2nd'
11730     =3:RETurn '3rd'
11740     =REMAINDER :RETurn n&'th'
11750   END SELect
11760 END DEFine
11765 :
11770 DEFine PROCedure setup_test
11780   TK2_EXT
11790   setup_read
11800   CLS
11810   setup_show #1
11820 END DEFine
11825 :
11830 setup_test
```

10290 to 10430: scr__lim% is a function which attempts to determine maximum window size by gradually reducing window dimensions and using FTEST to test if opening a window of such a size is possible on the current display.

10450 to 10500: Peek_W_Sys is a function which returns a word value peeked from a given system variable offset. It seeks to emulate the PEEK_W(!!addr) function found in Minerva and SMSQ/E versions of the operating system. Basically, if a Minerva or SMSQ or SMSQ/E system is detected, PEEK_W(!!addr) is used, whereas QDOS systems are handled by checking the system variables base address and adding the offset, then doing a PEEK_W from this address.

10520 to 10570: Peek_Sys is similar to Peek_W_Sys described above, but fetches a byte rather than a word value.

10590 to 10640: Peek_L_Sys is similar to Peek_W_Sys described above but returns a long word instead.

10650 to 11540: Setup_Read is the main procedure which reads the values and settings from the system. It starts off by establishing the operating system type in lines 10670 to 10920. The test for Minerva is VER$='JSL1' (and did you know that JSL1 comes from the forenames of the three partners in QView who first devised Minerva, Jonathan Oakley, Stuart McKnight and Laurence Reeves). The test for SMSQ is VER$='HBA'. Strictly speaking, these are the versions of SuperBASIC or SBASIC. The QDOS version is established with version$=VER$(1) if this is supported, i.e. on Minerva or SMSQ/E, otherwise the ROM version such as JM or JS is returned on older QDOS. If the BASIC concerned supports VER$(-2) this is used to determine the base address of the system variables, otherwise it's assumed to be decimal 163840 or (128+32)*1024 on a QDOS system as a standard QL does not support multiple or high resolution screens. The ROM language is assumed to be 'GB' for QDOS or Minerva, and if SBASIC is detected the LANGUAGE$ function is used instead.

10940: This line fetches the system variable byte at offset 161 (sys_ptyp) to check the processor type. This is encoded in the top 4 bits of the byte, so needs to be integer divided by 16 to get the value. In effect, the number returned is the x in the 680x0 processor type and the author multiplies the result by 10 to get the correct digit. Note that 68000 and 68008 are not distinguishable here. This makes this similar to the PROCESSOR extension in SBASIC.

10960: Checks the lower 5 bits of the machine type system variable (sys_mtyp) to provide a value similar to the MACHINE extension in SBASIC. This will include the test for Blitter or Hermes in bit 0, the chip being Hermes if not an Atari, or a Blitter if an Atari. Bit 0 being 0 means no Hermes or Blitter. See lines 11250 and 11260 for the "Atari test". In the case of the QLay emulator, the machine type 31 implies that QLay emulates a Hermes chip.

10980: Checks the display type, the top 3 bits in sys_mtyp. Returns a value similar to DISP_TYPE in SBASIC, but gives 6 for VGA on a QXL. DISP_SIZE gives a value of 0 as it assumes the QXL emulates the original QL display.

10990: Checks the current value of the display control register, in order to establish the display mode in line 11350 and display page in line 11360

11000 to 11040: Checks the frequency by checking the word value at offset 168 in the system variables. This gives 50 for 50Hz or 60 for 60Hz.

11050: Checks for the address of the top of RAM to establish how much memory is on this machine. The actual method of calculation is in line 11630.

11060: Checks for monitor display type. The value returned from the system variables is converted to something meaningful in lines 11290 to 11340.

11080 to 11230 converts the "hardware_code" value into the name of the type of machine, from Standard QL to QLay. Unknown values are simply stated as 'Unknown HW'. Note that with the release of type 17 hardware (Q40 or Q60) I have added line 11175 to cater for Q40 or Q60. Once Q40 or Q60 has been established, it's possible to distinguish them from the processor type (68040 or 68060).

11240 to 11280: Establishes the result of the test for Hermes or Blitter chips and groups the result depending on whether the machine is an Atari (Blitter with hardware_code 1,3,5,7,9) or not an Atari (i.e. Hermes with non-Atari hardware)

11290 to 11340: Establishes the monitor or TV display type, including if a 525 line USA TV mode or 625 line European TV mode for example.

11350: Display mode type is checked, but only works for 4 or 8 colour modes. To test for higher colour modes we'd need to test for SBASIC being present, possibly test for SMSQ/E versions 2.98 or higher (i.e. containing GD2 colour drivers) then use SBASIC's DMODE extension for mode number. SMSQ/E is currently the only system to support higher colour modes. These mode numbers can currently be as follows:

0,4,7,8,15,16,31,32,33,64

To adapt the program you may like to experiment with a line such as 11355 IF os$='SMSQE' THEN display_mode=DMODE

11360: This line checks the display page, i.e. whether the second screen is in use on a QL system or not (currently only supported by Minerva).

11370 to 11450: Convert display_code to a string describing the current display type.

11460 to 11520: Convert cpu_code to the name of the processor in question. Note that I have added line 11515 to the original listing to cater for 68060 systems such as Q60 and some Amiga systems which are powered by a 68060.

11550 to 11680: The SETUP_SHOW procedure displays the information gathered onto the screen.

11690 to 11760: A simple procedure to convert 1 to '1st', 2 to '2nd' and so on, in order to display the display page number (i.e. are we showing first or second QL screen for example)

11770 to 11820: Setup_Test is the procedure to call to start all this going!

If anyone has more material to add to this article I would be very pleased to receive it for publication. The more information of this type we can pass on to help our software writers, the better.
We have replies from Al Boehm and Daniele Terdina, which we will publish in the next issue.

A short Visit of XMenu - Part 2

Jérôme Grimbert

So let's go back to the explanation of PE in C:

Action routine of Loose Item

An action routine get three things and is expected to return another. It gets:
1. a pointer to the window definition (WM_wwork*)
2. a pointer to the loose item for which this function was called
3. a pointer to the window status (WM_wstat *)

And it should return a PE event (such as PT__CAN or AHIT_NORMAL).

One nice thing is that the status of the loose item has been already updated by the PE. Example for the list behavior:

```
static long ALIST(
 struct WM_wwork *wwk,
  struct WM_litm *li,
  struct WM_wstat *wst
)
{
  wm_ldraw(wwk,-1); /* redraw all changed
loose items */
  if (wst->kprs==K_DO)
  {
    return PT__CAN;
  }
  else
  {
    return AHIT_NORMAL;
  }
}
```

It's pretty simple: we redraw all the loose items, and according to the pressed key, returns either AHIT_NORMAL or PT__CAN. PT__CAN gets out of the pointer-reading function (wm_rptr() function), allowing to perform more things, such as cancelling a subwindow, or exiting a program. I remind you that a primary window is usually handled this way:

```
if (!wm_prpos(&ww,-1,-1))
{
   wm_wdraw(&ww);
   while (!wm_rptr(&ww))
   {
      if (ws->evnt & PT_CAN) break;
   };
}
wm_wrset(&ww);
```

whereas a secondary must use wm_pulld() instead of wm_prpos(). Of course, more PT_* events can be handled in the while loop. One last thing is that in order to be able to call the action function from the PE, we must use a wrapper. It a small piece of code:

```
static struct WM_action alist = { JSR,
wm_actli, ALIST };
```

In the loose item declaration, the .pact field is filled with &alist (pact stands for Pointer to Action routine). When the loose item is activated, the PE jumps at the .pact value, executing a JSR instruction to the wrapper wm_actli (action Loose Item) which then call your ALIST C function with the paramters correctly set for C code.

field of loose item

While I'm writing at the .pact field of the loose item (WM_litm *), I may as well give the other fields:

- **.xsize** : the horizontal size, in pixel
- **.ysize** : the vertical size, in pixel
- **.xorg** : the position of the left most pixel, relative to the window definition
- **.yorg** : the position of the top most pixel, relative to the window definition
- **.xjst** : horizontal justification, offset in pixel for graphics
- **.yjst** : vertical justification, offset in pixel for graphics
- **.type** : kind of item, TYP_TEXT or TYP_SPRITE or even a blob or a pattern (if you have provided the other part [pattern/blob] in the window definition)
- **.skey** : selection key, 0 if none
- **.item** : item number in the status structure, the place where the state of the item is stored.
- **.pact** : pointer to action routine

The size and origin are 'inside' the item border, which means that you must make a provision for the border width when placing the item. Two last points before ending about the loose items:

1. If you intend to use more than 40 loose items, you MUST update the definition of WM_wstat in the relevant system include file or you risk to get a memory corruption of your program (which, without MMU, will usually be unnoticed and provides very strange behavior sometimes!).
2. the horizontal justification for text item allows, by using a negative number, to underline some specific letters from the text. This is valid only for modern PE (if you still have a very old PE, it might not work). The underline letter should match the selection key.

Next to come: the information windows and how to use them!

Information windows

Re-drawing any information windows is done by calling wm_idraw(). The call takes two arguments:

- the windows definition (WM_wwork *)
- a bit field, on 32 bits.

The bit field is just a little strange: the bit number which matches the number of the information window must be clear if you want the redraw. You can not have more than 32 information windows, or then you won't be able to redraw the one you wants if the number is more than 32.

As a remark, anyway, more than 32 information windows is really too much. It might be ok to have 105 loose menu items, because each could

lead to a different action and there is only one object per loose menu item, but you can have many information objects in a single information window. So, there should no real need for too many information windows. The most I ever had was 13, in the sprite editor, and I really wasted half of them.

Being able to redraw only the information windows that need to be refreshed is a good thing, because it allows the unnecessary flickering and saves some time. Flickering can happen because the information windows hold a list of information objects. Each items in this list is draw in sequence, thus making 3D effects is pretty easy by drawing the biggest rectangle first, and then the smallers one, ending with the real content which can be a mix of text strings and sprite.

The fields of each WM_infw are:

- **xsize** : the horizontal size, in pixels
- **ysize** : the vertical size, in pixels
- **xorg** : the position of the left most part, in pixels, relative to the window (WM_wwork)
- **yorg** : the position of the top most part, in pixels, relative to the window (WM_wwork)
- **flag** : specify if it should be cleared and the size of the shadow
- **borw** : size of the border
- **borc** : colour of the border, on a byte, so palette might be possible but not true high colour. usually it's just the good old encoding of colours.
- **papr** : colour of the paper, on a byte too
- **pobj** : pointer to the object list

Information object

Each information object has the following field:

- **xsize** : the horizontal size, in pixels
- **ysize** : the vertical size, in pixels
- **xorg** : the position of the left most part, in pixels, relative to the information window (WM_infw)
- **yorg** : the position of the top most part, in pixels, relative to the information window (WM_infw)
- **type** : the kind of object, can be TYP_SPRITE, TYP_TEXT or even TYP_BLOB or TYP_PATT (for just a blob or pattern)!
- **spare** : a spare byte, should be 0.
- an union named attr alternating between:
  - for type TYPE_TEXT
    - **ink** : colour of the ink, on a single byte
    - **cwid** : the width of the characters (0 to 3, as for csize)
    - **chgt** : the height of the characters (0 to 1, as for csize)
  - for type TYP_BLOB and TYP_PATT
    - **comb** : the other part from a blob/pattern combination
- **pobj** : pointer to the object of the kind specified by type field.

Sharing the same object list among multiple information windows is possible, even if it does not help to compute the number of objects (well, all you have to do is just to count the number of objects that you want on the screen...).

And remember: if your objects are overlapping, the order in the list is important!

Next issue, we will deal with the application window.

Programming in Assembler - Part 10

Norman Dunbar

The Subroutine Library Revisited

First of all, apologies if you were expecting to see the next part of QLTdis in this issue, due to a large work load, Christmas and New Year, two weeks in the Dominican Republic and a serious bug that has come to light in one of the (unpublished) routines, I have a bit of fixing to do before I can turn you loose on the next exciting installment. Fear not, I hope to have it all fixed soon.
In part 9 and part 9/2 of the tutorials, I presented you with a hopfully useful library of subroutines and a small library handler to allow you to use them. Unfortunately, I owe all my faithful readers a huge apology - the code printed in parts 9 and 9/2 was so badly bug ridden I am seriously embarassed by the article.

I can only explain it away by noting that the file printed was an old version (one which I didn't know I still had) from my home machine. The version you should have seen, was the one I had on my Zip drive. Unfortunately, when I move articles from home to work - most of this is written in my lunch hours at work - I use the zip disc and I seem to have got a little out of sync at some point and trashed my good copies of the library. (Don't mention backups - we computer professionals never use them!)

So, as printed, the code is dangerous and unreliable. Personally, I'd be loath to use it myself. However, with a little bit of tidying up, we can use it again. Here goes, and again, my apologies for the mess.

PART 9 (Volume 6 Issue 3)

STR_REVERSE has, as its very first line, 'MOVE.L' rather than 'MOVEM.L'. Not a bad start, first line of first sub-routine and I got it wrong.

STR_INSERT makes a call to STR_APPEND if the value in D0.W is greater than the length of the string being inserted into. Unfortunately, STR_INSERT returns an error code in D0 while STR_APPEND does not. ROBERT NEWSON wrote a letter in Volume 6 Issue 4 on page 32 giving corrected code - I am grateful to Robert for noticing and fixing my feeble attempt.

PART 9/2 (Volume 6 Issue 4)

PRINT only preserves register A1. It should also preserve D1, D2 and D3 as these get trashed by the vector call UT_MTEXT. So, change these two lines:

```
MOVE.L A1,-(A7)
MOVE.L (A7)+,A1
```

to

```
MOVEM.L D1-D3/A1,-(A7)
MOVEM.L (A7)+,D1-D3/A1
```

LINE_FEED uses a couple of equates - 'linefeed' and 'infinite'. These are ones I use all the time, and have them in a separate file which is included by almost anything I'm working on. Unfortunately, I never mentioned this fact in the article, so attempting to assemble this code will not work unless you add the following:

```
LINEFEED EQU 10
INFINITE EQU -1
```

This can be done in a separate header file that you include, or simply change them in the library code I have given so that the instructions look like this:

```
MOVEQ  #10,D1
MOVEQ  #-1,D3
```

INPUT mentions in its outputs comments that A1 is the start of the buffer where the word count is'. This is not correct as A1 returns the address of the first character in the buffer. In addition, the line:

```
MOVEQ  #256,D2
```

should be rewritten as :

```
MOVE.W #256,D2
```

I have used the 'infinite' equate in the INPUT routine as well. Change this to '-1' or add an equate as explained above.

GEORGE GWILT has pointed out that as it stands, the input routine is 'ok' because the i_buffer storage will be word aligned, if any code or data gets inserted between the RTS and the definition of the buffer, it could cause the buffer to be located at an odd address which would cause a hang. To make sure that this never happens, change the definition of the buffer so that it is defines in WORDS rather then BYTES as follows:

```
I_BUFFER  DS.W  128+1
```

which will force word alignment regardless of what appears in the code before it. George has a few more observations too - you'll read them later.

JOB_HEADER also benefits from George's sharp eyes and mind. He has noticed that if the job name is an odd length, then the actual job name will have a 'bonus' character in it. This is caused by GWASL setting the 'prog_start' label to an even address - because it is an instruction address, and my code using a calculation to determine the length of the job name - as follows:

```
PROG_NAME  DC.W PROG_START-PROG_NAME-2
```

We need to change the code to look like the following instead - this will avoid the spare byte appearing in the job name.

```
PROG_NAME  DC.W PROG_NAME_END-PROG_NAME-2
    DC.B ''
PROG_NAME_END DS.B 0
PROG_START
```

Prog_start will still point to an even address, but the calculation now uses a 'dummy' end address for the name and so will get the length correct regardless of the 'oddity' of the name length.

MEM_ALLOC tests D0.L on return from the MT_ALCHP trap. It appears that the QDOS documentation sold by Jochen notes that this trap sometimes doesn't set D0 correctly, so there may not actually be an error code. This is a bit difficult to track down, and could lead to problems. My copy of Jochen's documentation doesn't mention this problem - but it does say that two error codes may be returned - Invalid job or Out of memory. I have always tested D0.L on return - just in case - but if George has more up to date docs than mine, then there may be a problem waiting to happen.

Perhaps we need more information on what circumstances the return in D0 is not correct so that we can work around it. In the meantime Geroge recommends taking the TST.L D0 out. (I'd prefer to leave that in - as I always have done. Difficult decision.)

MEM_DEALLOC also tests D0.L on return, however, this is not required because the MT_RECHP trap does not actually return an error code. To fix this routine, delete the TST.L D0 instruction and delete the comment which states that D0 is an error code.

SCR_MODE, **CLS**, **SCR_PAPER**, **SCR_INK**, **SCR_STRIP** all have the same problem. I have used 'MOVE' when I should have used 'MOVEM' when saving and restoring registers. All of those routines need to be fixed by changing 'MOVE.L' to 'MOVEM.L'.

SCR_MODE also has a serious error in it - twice! There are two instructions which are branches to the label SCRM_EXIT but SCRM_EXIT is in the wrong place. The end of the routine should read as follows:

```
SCRM_EXIT MOVEM.L (A7)+,D1-D2/D7/A3
    TST.L  D0
    RTS
```

And not as it currently does where SCRM_EXIT labels the RTS instruction.

SCR_PAPER, **SCR_INK** and **SCR_STRIP** appear to be suffering from 'cut and paste' fever in that each of them actually loads D0 with the SD_CLEAR code, rather than the correct trap code to set paper, ink or strip. At least I got the comments correct !

Change this line in each routine:

```
MOVEQ  #SD_CLEAR,D0
```

to one of the following:

```
MOVEQ  #SD_SETPA,D0   ; Set paper colour
MOVEQ  #SD_SETIN,D0   ; Set ink colour
MOVEQ  #SD_SETST,D0   ; Set strip colour
```

And finally, in the SuperBasic library manager utility, there is a missing semicolon on the third line down, it should read as follows:

```
INPUT 'Library name : '; LibraryName$
```

See you next time.

Drag & Drop with EasyPtr - Part 1

Wolfgang Uhlig

Recently there has been some discussion in the QL-Users-Group about the future of EasyPtr, which is still the only program suite for the QL to create pointer driven programs in (S)Basic. It seems that, despite the Quanta workshop some time ago, a lot of people still have great difficulties to program EasyPtr because of its cryptic manual. About 18 months ago I wrote a series of articles about programming in EasyPtr for the German QL-Today, and thought it might be helpful for some people to translate a small part of it.

So, what is it about, for whom is it, what can you learn and what do you need?

- The program to illustrate this article shows a list of 20 people. You can create a football team by dragging and dropping 11 of the names to another list. You can choose a week and print both the team and the week to ram. This is little point other than to learn something about EasyPtr-programming.
- You need to be able to create a menu with menu-items and application-windows in Easymenu_exe and to create a simple sprite with Easysprite_exe.
- You can learn how to display arrays in application-windows in different ways; how to calculate the size of an application-window; and how to deal with the contents of fields or

even move content from one application window to another.

- You first need: a small sprite symbolizing the thing that can be "dropped" after having "dragged" it. To make it simple, create an arrow or something like that. Second: we need a menu of about 350x250 pixels with 4 menu-items and three application-windows. The menu items should be: 1=ESC, 2=MOVE, 3=SLEEP, (2 and 3 set to automatic recognition) and a free menu item 4 with a name of your choice. I called it "showtime", and you will see why later. The size of the application-windows has a lot to do with their contents so before you create them, read what comes right after this introduction. For "well behaved" sprites it is important to store them in an appendix-file that can be lrespr'd when developing or attached to the code when compiling. Copy the file "ptrmen_cde" from your EasyPtr suite to a file with another name, I called it "projectmenus". (Attention: Never use the file "ptrmenr _cde"!) Then "append" both your menu and sprite to this file.

Okay let's start:

If you create application-windows for arrays, you have to take the following into account:

- The second dimension of the array, in our case the number of letters, must not extend beyond the width of the window. In addition, if you want to edit a string within a field, you have to add an input buffer of one letter.
- The height must not be smaller than a letter (or a number) is high. This seems obvious, but....
- if there are more fields than fit into the window, the window manager automatically creates **scrollbars and arrows which diminish** the height by 8 pixels on the top and bottom and by 14 pixels on the right side.

How do we calculate this?

On the QL letters and numbers are 6 pixels wide. If we are planning to create an array of for example 20 names with a maximum of 18 letters, thus:

```
DIM names$(19,18)
```

We count:
18 letters x 6 pixels = 108 + 6 pixels buffer + 14 pixels for the scrollbar = 128 pixels

So please create two application-windows with a width of 128 pixels. The height is less important here, because, as I have already said, scrollbars are automatically created, when the array is too long for the window. For a better overview, however, don't make them too small.

The third application-window is meant to be very small and to show only one number out of 52, so height as well as width are of importance. With

```
DIM weeknumber$(51,2)
```

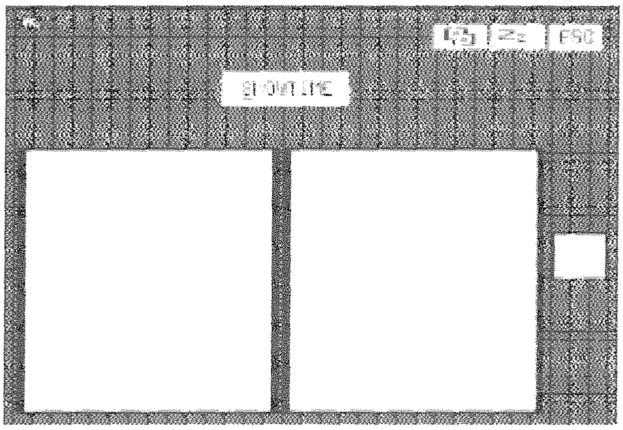
we also create this array as a string-array as only string-arrays are possible in EasyPtr.

A QL-letter is 9 pixels high and the scroll-arrows take 8 pixels, so we get for the height:
9 pixels + 8 pixels top and 8 pixels bottom = 25 pixels

and for the width
2 numbers x 6 pixels = 12 + 14 pixels for the scrollbar =26 pixels

Here we need NOT add a buffer of 6 pixels, because the numbers are not to be edited, but only displayed (Strictly speaking this was also the case for the other two application-windows, but we need to learn the principles...). An application-window of 26x25 is therefore sufficient to show numbers up to 99.

Please complete your menu now so that it looks similar to the one shown below. The program code will refer to the application-window on the left side as number one, the other big one number two and the small one number three.



At last we can now start with the program.
You can start to type it in now, and I will give a detailed explanation of the program in part 2 - in the next issue.

```
100 REMark ***** a small EasyPtr-practise
110 :
120 LRESPR win1_projectmenus
130 :
140 OPEN #3,con_
150 MDRAW #3,1
160 INIT
170 REMark **********************************************
180 REMark ***** start of the main program ***********
190 :
200 REPeat main
210 key=MCALL(#3,key,0)
220 SELect ON key
230  =-1:MCLEAR #3:CLOSE #3:STOP
240  =-4:SHOWTIME
250  =2^16 TO 2^32:
260           awnum=key
270           position=MAWNUM(#3,awnum)-1
280           IF awnum=3
290            NEXT main
300           ELSE
310            DRAG_DROP
320           END IF
330  =REMAINDER :NEXT main
340 END SELect
350 END REPeat main
360 REMark ***** end of the main program *************
370 REMark ***********************************************
380 DEFine PROCedure INIT
390 REMark ***** define/set variables for drag&drop
400 drag%=0:buffer$=""
410 :
420 REMark ***** create an array for 20 names
430 DIM names$(19,18)
440 RESTORE
450 FOR i%=0 TO 19
460  READ a$
470  names$(i%)=a$
480 END FOR i%
490 DATA "Roy","Darren","Tony","Dilwyn"
500 DATA "Jochen","Wolfgang","Marcel","Dietrich"
510 DATA "Al","Dave","Phoebus","Nasta"
520 DATA "Thierry","Jerome","Francois","Claude"
530 DATA "Sjef","Per","Jonathan","Derek"
540 :
550 REMark ***** create an array for the selected players
560 REMark ***** which is empty at the moment:
570 DIM selection$(10,18)
580 :
590 REMark ***** an array for 52 weeks
600 DIM weeknumber$(51,2)
610 FOR i%=0 TO 51:weeknumber$(i%)=i%+1:
620 :
630 REMark ***** now draw all arrays:
640 MAWDRAW #3,1,names$
650 MAWDRAW #3,2,selection$
660 MAWDRAW #3,3,weeknumber$
670 END DEFine INIT
680 :
690 DEFine PROCedure DRAG_DROP
700 IF drag%
```

```
710  IF MTEXT$(#3,key)<>""
720   NEXT main
730  ELSE
740   IF awnum=1
750    names$(position)=buffer$
760   ELSE
770    selection$(position)=buffer$
780   END IF
790   MAWITEM #3,key,,buffer$
800   buffer$=""
810   drag%=NOT(drag%)
820   SPRS #3,0
830  END IF
840 ELSE
850  IF MTEXT$(#3,key)<>""
860   IF awnum=1
870    buffer$=names$(position)
880    names$(position)=""
890   ELSE
900    buffer$=selection$(position)
910    selection$(position)=""
920   END IF
930   MAWITEM #3,key,,""
940   drag%=NOT(drag%)
950   SPRS #3,1
960  ELSE
970   NEXT main
980  END IF
990 END IF
1000 END DEFine DRAG_DROP
1010 :
1020 DEFine PROCedure SHOWTIME
1030 LOCal number
1040 number=MAWNUM(#3\3)
1050 open_over #4,ram1_showtime
1060  print #4,"In week "&number
1070  print #4,"the following players were selected:"
1080  print #4,selection$
1090 close #4
1100 END DEFine SHOWTIME
```

The Extended Environment in SBASIC: Programming with QPTR

Wolfgang Lenerz

This is the new Guide to using QPTR in SuperBASIC. The purpose hereof is to enable you to program the "Extended Environment" - so called the (Pointer Environment) very easily with the QPTR extensions (which you must obtain seperately). Contrary to what a first impression may let you believe, the Extended Environment, and QPTR at the same time, are not complicated or difficult, but just complex, notably because there are so many new concepts to assimilate at once. But it is actually sufficient to know and respect its "philosophy" to see - and understand - the logic behind it.

I sincerely hope this Guide will be useful to you.

Introduction

This is an explanation of the concepts and keywords needed to *program* applications using the QPTR SuperBASIC extensions. For some aspects, we will use examples derived from QPAC II, it is thus hoped that the reader is familiar with this software...

Before starting on the course proper, some terms might need an explanation:

The **Extended Environment** essentially is just a "new" method to interact with the user of an application:
Interaction means, on the one hand, display of information on the screen (in windows) and, on the other hand, obtaining the user's response to this information (often, but not always, through a 'pointer'). For example, a file copier displays information (the name of files on a disk) and obtains the user's response (i.e. selecting which files to copy). The Extended Environment (which I'll abbreviate as EE from here on) can handle that aspect of a program, but the rest of the program will remain (nearly) unchanged: in the example just used, it is still you, the programmer who will have to program the copying operation itself.

An **application** is simple a program.

A window is said to be **managed** when it is part of an application written specifically to take advantage of the facilities offered by the EE: QPAC II has managed windows, QUILL has windows that are not managed.

The EE changes not only several aspects of the QL's windowing system, but also the QL's multitasking. Here, the concept of a window is enlarged to mean not only the means through which an application will communicate with the user, but also the means to determine whether an application will multitask or will be suspended.

The best way to understand that is to imagine that, for the EE, all window are "stacked" on a pile (one considers that an application has but one main window). The window that is on top of the pile is that which is entirely visible on the screen. This window is said to be **unlocked** which means that it will accept input if you type something in it and if the mouse pointer is over it. If you now hit CTRL + C, then the window on the top of the pile will get transferred to the bottum of the pile, and the window that was just under that one will be on top of the pile. Now, if the pointer is in that window anything you type (or any click of the mouse) will be directed to that window and is thus taken into account by the application to which that window belongs. The window on top of the pile will be called the upper window. The other windows, which are underneath it, do not accept keystrokes. It is then said that thay are **locked**.

It is possible for two or more windows to be on top of the pile at the same time, and to be visible entirely (if they are small enough...). Both windows will then be unlocked. However, if anything is typed on the keyboard, the keystrokes thus generated go only to the window in which one can see the pointer.

The concept of ***locked windows*** is important: An application whose window is locked will be suspended (i.e. it stops working) until its window becomes unlocked IF this application either attempts to write to the screen or is waiting for user input. Example: You are working in Abacus, and ask it to recalculate a large speadsheet. As soon as Abacus starts to do that, you switch to Basic. Abacus will continue to work on the spreadsheet, until it has to display the recalculated sheet. Then it will stop cold, waiting for you to switch back to it (thus unlocking it window). Until then, Abacus is suspended.

The word **pointer** can have two meanings: first of all it can mean the concept of a pointer as used in all programmming languages, i.e. a variable pointing to something. Also, it can mean a pointer (cursor) on the screen, moved about by the mouse or cursor keys. One doesn't generally use the word "cursor" because that normally ony means a rectagular square (blinking or not), whereas a pointer can have about any shape you desire. Normally, it should be quite clear from the context which meaning of the word pointer is used, without any possibility of confusion.

The mouse pointer can be used to "hit" objects or "do" these objects: a "Hit" is either a click of the left mouse button or tapping the space bar. A "Do" is either a click with the right mouse button, or tapping the ENTER key.

An application will have a main (or "***primary***") window through which it communicates with the user. Generally, this window will be divided into "***sub-windows***". These sub-windows are but subdivisions of the main window. Thus, in QPAC II, for example, the primary window of the "Files" menu is the entire window visible. The part of the window which displays the file names is a sub-window.
Some sub-windows are a bit special in that they can have 'objects' (such as the file names in the QPAC II files menu). The state of the objects can change when hit or "done" and can even produce an action. The sub-windows are called ***application sub-windows***.

Programs using QPTR can also be **compiled**. However, you have to use the QLiberation Software's QLiberator for this, as the "Turbo" compiler cannot cope with functions and commands which return changed parameters (even thought this is explicitly foreseen for Basic keywords). As an importnt number of QPTR keywords use this feature, programs written with them cannot be compiled with "Turbo".

A program written for the EE will most likely follow the following procedure:

- Definition of window(s)
- Display of windows onscreen
- Waiting for user input
- Act on user input
- (perhaps) Re-define windows and display it
- Wait for user input etc...

This is in fact not far from 'classical' programming: QUILL doesn't do anthing else than display its windows, wait for user input, act on that etc...

Each of these stages will be discussed. The most difficult and important is the first stage, the definition of the window.

Part One: Defining the Window

We wish of course, to define windows which are managed. To this end, there are rules to be obeyed, the definition must be made in a determined manner, which may seem complicated at first. To obtain this global definition of the window, there are several **levels of definition** through which you will successively have to pass: You must first define the main window, then the different sub-windows and lists.

I - LEVEL I: Definition of the primary window

A - Some new concepts

A certain number of new concepts must be set out before we can examine the new keywords.

1) The primary window

The primary window of an application is terribly important. Put simply, it is the first window to be opened for an application - but it determines the graphical aspect of the entire application. This is why it is called the ***primary window:*** Primary not only because it is the first to be opened, but also because it primes all the others.

The primary window is paramount: your application is not allowed to open any other window outside the primay window!

2) The sub-windows

As we shall see, and as was already mentioned, the primary window itself is generally broken down into sub-windows. There are different kinds of sub-windows, each doing it's own bit. NONE of the sub-windows' sizes may exceed the size of the primary window, they must all be opened within their primary. Even if you attempted otherwise, the EE would not let you (!).

There are three types of sub-windows:

- The ***information sub_windows*** which just display some information, as their name suggests.
- The ***application sub-windows*** - they can be so diverse that it is difficult to give a precise description.
- The ***menu sub-windows***: these are a special case of application sub-window, containing "objects".

3) Menu items

In addition to sub-windows, primary windows may also have "loose menu items" (sometimes also called simply "menu items"). These can change state or produce an action when hit or "done", and, if the pointer moves over them, a border is drawn around them.

To understand these three components, let's look at the QPAC II "Files" menu:
We can see the first sub-window, containing the file names, at a first glance. This is a menu sub-window (the file names are its objects).
We also notice the menu items, such as F3: Commands, F5 All etc...
The "stripes" around the device name, are drawn within an information sub-window. Likewise, the data on the device (free sectors/ total sectors) are displyed within an information sub-window.
It should be noted that none of these sub-windows is a "window" in the QL sense (i.e. having its own SCR or CON channel), even if they behave like such. This can be seen from the "Channels" menu: the "Files" menu has but one screen channel open...

4) Secondary windows

Sometimes it is necessary to have additional windows which are true QL windows, with their own channel. These will generally be secondary windows.

RWAP SOFTWARE

QL Cash Trader v3.7 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT returns, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of the current tax year rules.

Q-Help v1.06 £10
Q-Index v1.05 £5

Q-Help: on-screen help for SuperBASIC commands, including TK2, Turbo Toolkit, SMSQ/E and PD toolkits. Can be used to add help to your own programs - simply produce ASCII text for each help page, add an index and Q-Help automatically cross-references and displays the links.

The PD toolkits referred to are available for £2.

Q-Index: The SuperBASIC index supplied with the Reference Manual - enter a topic such as 'screen resolution' and find out the commands which relate. Launch Q-Help for further info on the chosen command.

Sidewriter v1.08 £10

Produce landscape printouts of Easel/Qspread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix to laser printers. A most useful utility by Dilwyn Jones - you know it must be easy to use.

ProForma ESC/P2 Drivers v1.04 £8

New improved colour and monochrome printe
drivers, providing up to 720dpi for all progr
written for use with ProWesS, such as LineDesi
Paragraph. Works on all Epson inkjet print
support binary mode compression (740, 8
models at least). 1440 dpi to follow.

QL Genealogist v3.26
Genealogy For Windo

Store your family tree f s
with details of their p ll of
those links build u layout.
Text files and linked to
individuals a , making this
the perfec y of your family.
QL versi and QMenu as well
as a ale and female trees.
Samp y since 1066 included.
PC ve - enter the details as they
appear i nd it generates the tree from
these. QL EDCOM can be transferred to the
PC version. e to latest PC version (v5.21) for £8
Both program easy to use and complete with a step by step tutorial.

** QL USERS upgrade to PC version for £25 ONLY **

Image D v1.03 £10

Produce graphical representations of 3D objects - view them as wireframe, hidden line and shaded. Perspective and magnification can be controlled and views can be saved to file for subsequent printing. Multiple objects can be defined and positioned relative to each other. Simple to use yet produces excellent results.

SBASIC/SuperBASIC Reference Manual £40
Updates £6 each, £10 for 2 (Current Version - Rel 4)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on each of the different QLs, plus SMSQ/e, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.

This book is ideal for all QL users and is kept up to date with regular updates.

Orders are currently being taken for the next print run of this popular tome.

(Note: Price for the book does not include post & packing).

QL Cosmos v2.04 £5

Ever wondered what the stars in the sky looked like 100
years ago? Or, maybe you want to learn the constellations
and names of what you see in the sky. This is the progra
for you - generates pictures of the stars and planets for
given place or time and provides details on these o
Includes Halley's Comet, the Moon and the Sol
planets.

Q-Route v2.00
Upgrade from v1.xx

The latest version of this popu
Find the quickest route or th y
two places, using roads. A lable
for this program (see els rogram
is easy and quick to u wn places
and roads to the m

Flashbac nly) £5

The ultima mely fast and flexible,
easy to he latest versions of the
QL o ntained. A report module is
i at output in any way, including
 only available as an upgrade from
 original still available from Sector

den v3.08 £10
MKII v2.03 £8
rawn v2.01 £8
rrorday v3.1 £8
West v2.00 £5
The Lost Kingdom of Zkul v2.01 £5

A wealth of QL adventures - mainly text only.

Save the Galaxy from the ambitions of the evil dictator Nemesis.

Battle against werewolves and dracula look-alikes on a Hammer Horror set in the comical Horroday.

Take the part of a prawn with a hangover, lost in a strange land in the hilarious Prawn.

Solve a bank-robbery by fighting the bad guys and collecting the loot in real-time old West.

Battle countless dwarves in the atmospheric Lost Kingdom of Zkul.

Return to Eden is a massive adventure over 3 disks with colourful graphics - control 3 characters in their quest to find the missing Prince.

All six adventures are available together for only £25.

* SBASIC Reference Manual now Rel 4 - covers colour drivers *
New versions of Q-Help, Q-Index and ESC/P2 ProWesS printer drivers
*** Watch out for QWord - Coming soon ***

A range of games to keep both the young and the young at heart amused. Some are old favourites, like Golf and a pub quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for? All 6 programs only £28.

lf v5.20	£8
r II v2.07	£5
er II v2.00	£5
ne v1.2	£5
strike v1.5	£5
F ghtdeck v1.05	£10

The latest maps for Q-Route. Maps o
areas of Britain have been created by
them out of Big Britain Map - they will us
memory and can contain more detail. A
covered: Scotland, NE England, NW Englan
S&W Yorkshire, Wales & Derbyshire, London area
and South England.
Latest version of Q-Route is recommended.

Britain.map v1.11	£2
BIG Britain Map (needs 2MB) v2.03	£5
Various Britain Area Maps (ask for details)	£2 ea.
Ireland Map v1.00	£5
Belgium Map v1.01	£2
Catalonia Map v1.03	£2

D-Day MKII v3.04	£10
Grey Wolf v1.8	£8
War In The East MKII v1.24 (Upgrade Only)	£5

For the gaming enthusiast - D-Day is a classic table top wargame for one or two players - you control either the Allies or the Axis forces during WWII. With the ability to define your own army set ups and a choice of 4 different scenarios, this should keep you entertained for a while.

Grey Wolf is a graphical simulation of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

RWAP Software, 7 Common Road,
Kinsley, Pontefract, West Yorkshire
WF9 5JR
TEL: 01977 614299

Cheques in £sterling
payable to 'R.Mellor'

* Also known as Trading Accounts

Secondary windows are defined, and behave, exactly like primary windows (i.e. they have their own sub-windows, menu items etc...) BUT these secondary windows are all confined within the primary window whose size they may not exceed. Simply put, an application may not display anything outside its primary (but it is possible to make the primay bigger, if need be).
An example of a secondary window: in the QPAC II Files window, hit F3. This opens another window, which itself has menu items. This other window is a secondary window, it has its own CON_ channel, as you can see when checking through "Channels".
It is of course possible to open a new secondary window within a first secondary window (no, they're not called tertiary windows...). This can be useful if you wish to have a cascade of menus: a first menu leads to a second one, which in turn leads to another etc... (it is not, however, considered to be good programming style to use too many cascading menus).
Whilst any secondary window is, of course, limited to the size of the primary, a secondary window within another secondary window is NOT limited to the size of the first secondary window - else, successive menus would have to get progressively smaller!
In brief, an EE application has two kinds of windows: one primary window (possibly with sub-windows) and, possibly, one or several secondary windows. Each may have its own loose menu, and sub-windows.
The difference between a menu, whether it is a loose menu or the objects in a menu sub-window on the one hand, and a sub-window,on the other hand, is the fact that clicking on an item in a menu will lead to a result. This may just be to select the item (e.g. F4 - view in the QPAC II Files menu) or lead to some kind of action (e.g. F3 in the QPAC II Files menu). Clicking on a sub-window in itself generally produces no results (there is one exception to which we will come later)..

5) The working definition

To construct a primary window, you will need to build up a "**working definition**" of this window. Let's take an example with "normal" SuperBASIC. You can open a window just by typing: "OPEN#3,con_". You have then opened a window. However, to really define this window, you would then define its size and position (WINDOW#3,x,y,z,p), its colours (border, paper, ink) etc... Thus you will build up an exact definition of your window, with all your parameters.

Likewise, in the EE, you make a definition of the window according to your parameters. Here, however, this defintion, i.e. the "working definition" is more complex and it is compulsory - you cannot do without it.

B - Making the Working Definition

The working definition of the primary window is built up by the following function: **MK_WDEF** (MaKe Working DEFinition).

```
workdef= MK_WDEF (wdef%,wattr%,wptr,ltab,
inftab,apptab)
```

"workdef" then becomes a pointer to the working definition of the window. The parameters to this function are as follows:

-› * **wdef%** is an array containing the "physical" definition of the window.
In other words, it is a 4 element integer array (DIM wdef%(3)). Its elements are, in this order:
- window x size
- window y size
- x position of pointer when the window is drawn
- y position of pointer when window is drawn.

The pointer position is given as the number of pixels starting from the upper left hand corner of the window, which is considered to be at coordinates (0,0).

-› * **wattr%** is an array containing the window "attributes". These "attributes" are simply the following: window paper (& strip) colour, size and colour of the window border and size of the shadow beneath the window, in the following order:
- size of shadow
- size of border
- colour of border
- paper colour

So, there again, this is an integer array with 4 elements (DIM wattr%(3)).
The last three parameters should be clear to anyone concerned. The "size of the shadow" is given in pixels (but is multiplied by 2 by the software, to have even numbers). The shadow counts for the size of the window: On a normal QL, you could not have a window 512x256 pixels wide plus a shadow, this would make the window too large. A shadow size of 2 is generally thought to be sufficient.

-› * **wptr, ltab, inftab, apptab** are level II "pointers" (i.e. they are explained in level II):
  - wptr is a pointer (generally obtained by SPRSP) towards a sprite definition.
  - ltab is a pointer to a loose menu items list, as returned by the MK_LIL function: ltab=MK_LIL (level II parameters).
  - inftab is a pointer towards an information sub-window list, as returned by the MK_IWL function:
    inftab=MK_IWL (level II parameters).
  - apptab is a pointer towards an application sub-window list, as returned by the MK_AWL function:
    apptab=MK_AWL(Level II parameters).

Each of the last 4 pointers may be set to 0. In this case, it is considered that the list to which it points does not exist: if inftab = 0, there are no information sub-windows.

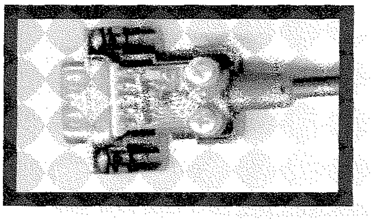
ATTENTION: It is important to respect the types of variables: if a variable is expected to be an integer, or an integer array, the variable MUST be of the correct type. Else, at best, the function in which it is used will give up with an error, at worst very bizzarre things may happen...

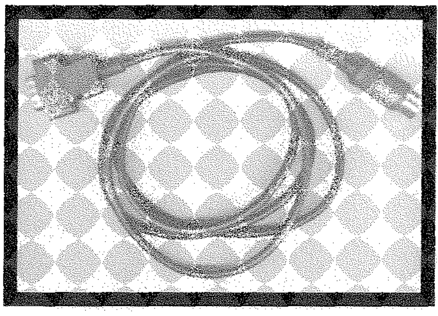
We'll continue with level II functions in the next instalment of this series.

Sinclair QL CSYNC Inverter

Marcel Flipse

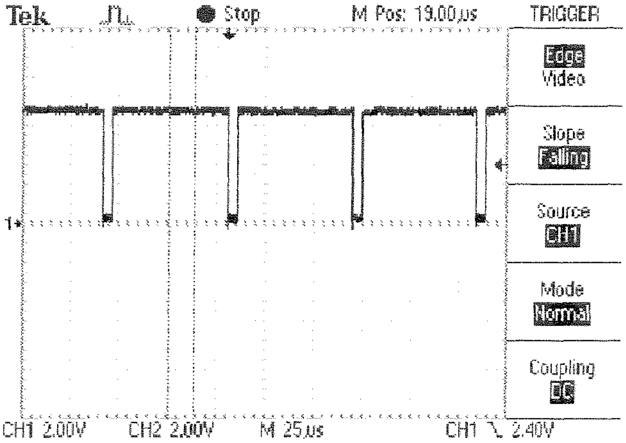
The QL cannot be connected to a CGA monitor directly. This is because the QL has an active-low Csync pin. A CGA monitor expects an active-high signal. This document shows how to make a very little circuit board, which inverts the Csync pin. No additional power supply is needed. The PCB is small enough to fit inside a DB-9 connector.





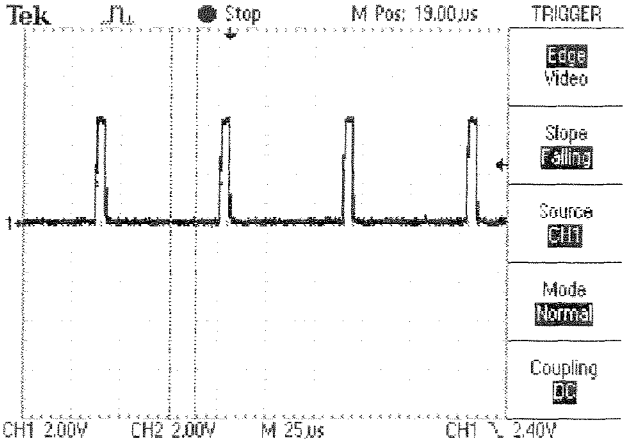
Here are some pictures of the cable.

The circuit is straightforward. The Csync signal is fed through a single gate NAND, which acts as the inverter. The NAND gate is powered by the Csync signal itself. Energy is stored in a 10uF tantalum capacitor, to buffer the time the Csync signal is low. A 100 ohm resistor is added to limit the inrush current during power-up. Some additional resistors and SOT-23 transient suppressors are added for extra protection of the QL.



Tek ● Stop M Pos: 19.00μs TRIGGER
Edge Video
Slope Falling
Source CH1
Mode Normal
Coupling DC
CH1 2.00V CH2 2.00V M 25μs CH1 ↘ 2.40V

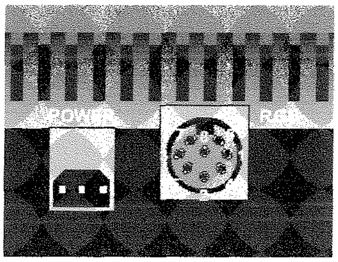
Here you can see the 'original' Csync signal, measured directly at the 8-pin DIN connector at the rear of the QL.



Tek ● Stop M Pos: 19.00μs TRIGGER
Edge Video
Slope Falling
Source CH1
Mode Normal
Coupling DC
CH1 2.00V CH2 2.00V M 25μs CH1 ↘ 2.40V

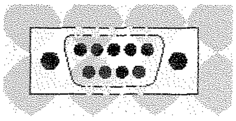
Inverted signal

QL monitor pinout



1	PAL	Composite PAL
2	GND	Ground
3	Video	Composite Mono Video
4	Csync	Composite Sync
5	Vsync	Vertical Sync
6	Green	Color Signal
7	Red	Color Signal
8	Blue	Color Signal

CGA DB-9 pinout



1	GND	Ground
2	GND	Ground
3	Red	Color Signal
4	Green	Color Signal
5	Blue	Color Signal
6	-	
7	-	
8	Hsync	(Inverter QL-Csync)
9	Vsync	Vertical sync

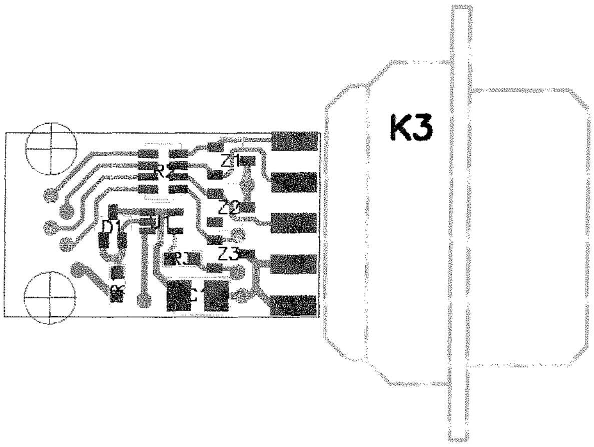
The schematic drawing is shown on the following page. The following components are needed:

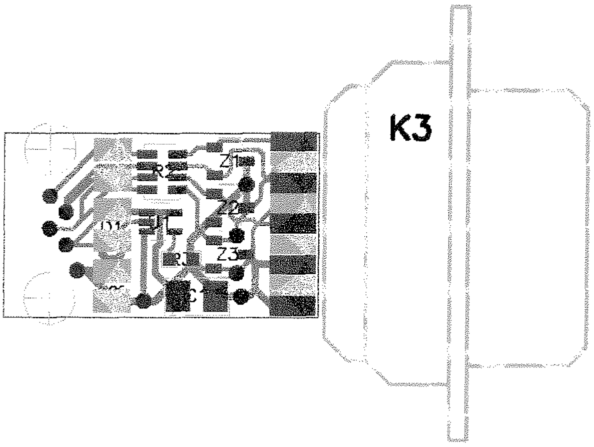
Qty	RefDes	Component	Value	Description	PatternName
1	U1	74AHC1G00GW		Single Gate NAND	SOT353
1	R2	ARC241	100R	Resistor Array	ARC241
1	D1	BAV70W		Dual Diode	SOT323
1	C1	CAP_SIZEA	10uF	Tantalum Cap.	SIZE_A
1	K3	DB9M_EDGE		SubD Connector	DB9EDGEM
3	Z1, Z2, Z3	MMBZ6V2ALT1	6.2V	Transient Abs.	SOT23
2	R1, R3	RES_0603	100R	Resistor	0603

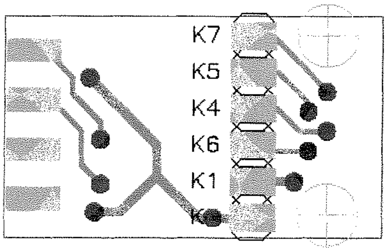
Additional stuff:
6 core cable (screened if possible).
8 pin QL style DIN connector.
Sub-D hood.
ADT09 from Assmann, www.assmann.com, see picture on first page.
2 Washers, which are below the two screws inside the sub-D hood. See picture.
The circuitboard itself.

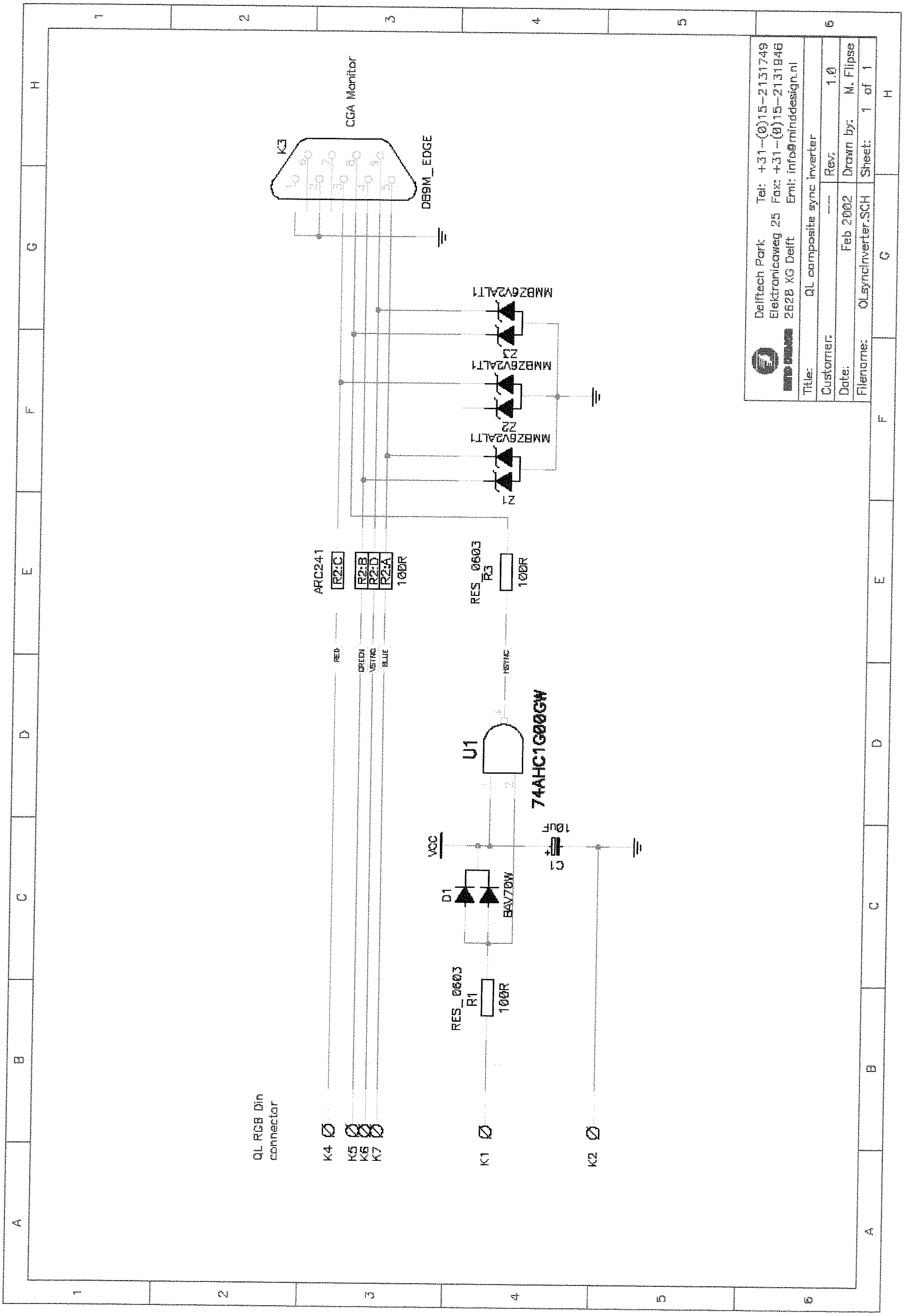
Mail me (**mflipseNOSPAM@zonnet.nl**) if you need any gerberfiles or additional info.

Don't fry your QL building this circuit! Good luck!








CGA Monitor
K3
DB9M_EDGE
MMBZ6V2ALT1
Z3
MMBZ6V2ALT1
Z2
MMBZ6V2ALT1
Z1
ARC241
R2:C
R2:B
R2:D
R2:A
100R
RES_0603
R3
100R
RED
GREEN
VSYNC
BLUE
HSYNC
U1
74AHC1G00GW
VCC
C1
10uF
D1
BAV70W
RES_0603
R1
100R
QL RGB Din connector
K4
K5
K6
K7
K1
K2
Delftech Park
Elektronicaweg 25
2628 XG Delft
Tel: +31-(0)15-2131749
Fax: +31-(0)15-2131946
Eml: info@minddesign.nl
Title: QL composite sync inverter
Customer: ---
Rev: 1.0
Date: Feb 2002
Drawn by: M. Flipse
Filename: QLsyncInverter.SCH
Sheet: 1 of 1
1
2
3
4
5
6
A
B
C
D
E
F
G
H

Aircraft shining after Sunset

Alex Wells

Have you noticed, a while after sunset, that a plane flying overhead glints in its own sunlight? I wondered whether I could work out how long this effect might last.

Perhaps I could use one of my 'Astro' QL programs (in the Quanta library, based on 'Astronomy with your personal computer', by Peter Duffett-Smith) to do some useful (!) sums. Some rough assumptions are needed:

```
nominal
  Radius of Earth = 3964 miles
assume
  Altitude of airplane = 7 miles (about 37000 ft.)
then - angle up from airplane's horizon to airplane
   = ACOS ( 3964/3964 + 7 ) rad.
                            = 3.4 deg.
```

The QL's "ACOS" is the inverse trigonometric function of "Cosine"

Use the procedure 'HsunRS' (sun Rise & Set) in the 'Astro_MsunS' program.
Calculate sunset time at home (2 deg. west, 53 deg. north)
Calculate sunset time 3.4 deg. further west.
This is the time for sunset at the airplane overhead.
No, it is not - this calc. is true on the equator at the equinoxes, but it gives an order of magnitude of the effect of the aircraft altitude!

Subtract the two sunset times to give an estimate of how long after my sunset on the ground it would be for the shining airplane effect to be possible at the given altitude.

The output of the program follows:

MsunS 2002 Jul 05 22:03:32

Sunrise and sunset

```
Geographical longitude { D, M, S; W neg. } ..  -2  0  0
Geographical latitude  { D, M, S; S neg. } ..  53  0  0
Daylight saving .... { H ahead of zone t } ..  1
Time zone ....... { hours; West negative } ..  0

Calendar date ............... { D, M, Y } ..  5  7  2002
```

Circumstances of sunrise

```
Universal time ......... { H, M, S } .....    3 49 27.40
Local civil time ....... { H, M, S } .....    4 49 27.40

Azimuth ... { D, M, S; zero is North } ...  + 48 26  4.66
```

Circumstances of sunset

```
Universal time ......... { H, M, S } .....   20 35  3.61
Local civil time ....... { H, M, S } .....   21 35  3.61

Azimuth ... { D, M, S; zero is North } ...  +311 25 49.88
```

Sunrise and sunset

```
Geographical longitude { D, M, S; W neg. } ..  -5  24  0
Geographical latitude  { D, M, S; S neg. } ..  53  0  0
Daylight saving .... { H ahead of zone t } ..  1
Time zone ....... { hours; West negative } ..  0
```

Circumstances of sunrise

```
Universal time ......... { H, M, S } .....    4  3  3.90
Local civil time ....... { H, M, S } .....    5  3  3.90

Azimuth ... { D, M, S; zero is North } ...  + 48 26 11.07
```

Circumstances of sunset

```
Universal time ......... { H, M, S } .....   20 48 39.28
Local civil time ....... { H, M, S } .....   21 48 39.28

Azimuth ... { D, M, S; zero is North } ...  +311 25 43.16
```

The result is:

my local sunset is at 9.35 pm and an estimated airplane sunset occurs after 9.48 pm.

So I should be able to see my plane still glinting over 13 minutes after sunset, if the assumptions are good.

Editor's note: this article is a little specialised, so I have included Alex's email address
alex_wells@onetel.net.uk
in case readers wish to correspond with him about this article. Don't forget that the 'Astro' program he refers to is available from the Quanta software library.

Letter-Box

Marcel Kilgus writes:
Dilwyn Jones wrote in the last issue "Despite this, QPC2 has some way to go before it can match the 20kHz stereo audio from a Q60 for example". I'd just like to point out that originally I planned QPC2 v3.02 to have samples sound support. The problem is just that the SSSS was specifically designed with the simple Qx0 sound system in mind and ironically it is quite difficult to emulate it with the more sophisticated PC sound chips. Doing it in a way that it sort of works in most cases is not very difficult, but for that I'm too much of a perfectionist. So far I am not sure how I will solve this issue. At least one option would be to design a new sound system interface (that could also provide for sampling rates other than 20kHz, like the more common 22050Hz rate) and only provide SSSS support through an emulation layer. Anyways, stay tuned for more news on this subject in a future QL-Today issue.

Wolfgang Uhlig writes:
Reading the last QL-Today I felt a little bit irritated by the way Dilwyn Jones promotes the Q60. I can accept his enthusiasm about it, but I don't think that the EDITORIAL is the right place for what sounds more like an advertisement to me. In the article about the QUANTA workshop he writes something which in my opinion is nonsense: "Despite this (the implementation of BEEP in QPC) QPC2 has some way to go before it can match the 20kHz stereo audio from a Q60..." As far as I can see there is absolutely no necessity for QPC to have a sound system. QPC runs per definitionem on a PC and the user has therefore easy access to state of the art sound systems and applications. Thanks to Marcel we can start sound applications even from within QPC! Reading the discussions on the QL-User group I learnt (and Dilwyn could, too) that the 20KHz Q60 sound is actually an inferior system and Marcel and Nasta are looking for something that meets modern standards. So what is this statement of Dilwyn about? Can't we stop promoting one system by running the other one down?

Swiss QL Meetings

Urs König

Today (Sat., 27th of July 2002) we held the 212th (approx.) meeting of the Sinclair User Club Schweiz (SUCS). For the first time in years I had time to join and meet good old friends like Markus Dettwiler, Jonathan Dent and his wife Lisbeth, Rudolf Rindlisbacher and Rolf Schneider.
Jonathan recovered well from his bike accident so far, he started back at work this week.
Jonathan and myself had our notebooks with us. He showed how he uses RedHat Linux and UQLX to develop his TCP/IP-stuff and how he manages to use "server" (Linux) and "client" UQLX at once on one CPU. He uses two USB-ports with two USB2COM-Converters as the native COM-ports on his notebook do not work properly.
I showed 5 short MPEG movies and some 50+ pictures I've made at the US QL-show and my trip to Niagara Falls in June 2002.
I found a place (angelfire) to host all my pictures and movies of 1995 US show, QL2000 and the 2002 US show. I hope to upload them soon.

Small Ads

We have decided to allow for free small ads - but we reserve the right to limit the number of small ads as space permits, or not publish them at all if we find it is not QL-related. Please honour the deadlines (see page 2).

For Sale

Sinclair QL with AZERTY (French) Keyboard, bought in 1984, equipped with a 2MB Extension including TK2 and floppy disk controller. 200V Power supply, Monitor (about 15") and some software like Text87 Plus 4, QPAC1, QPAC2, LineDesign etc. ... and original documentation.

Sadi de Luna, 25 Rue Pradier, 75019 Paris, France. EMail: said.deluna@wanadoo.fr

For Sale

AT cased system, comprising Aurora with Super Gold Card, Qubide, Superhermes, 3.7GB HDD, Keyboard, IBM Monitor and Mouse. The system can be seen on **www.macnamaras.com** and then under 'Whats New'. Price £250.
Contact **Mike MacNamara** for further details on **mike@macnamaras.com** or phone 01383-824494

BYTS OF WOOD

SAW POINTS OFFCUTS AND SNIPPETS

I was idly flipping through one the PC trade magazines at work and I learned that it was seventeen years ago that Clive Sinclair first began to admit that his company was in trouble. A vast outlay on a kind of electric canoe with wheels which no-one, not even the most die hard anorak, would want to be caught around town in was cited as the cause of the company's woes although the QL did get a mention in an aside which noted that the business community had failed to take to the new computer. I began to think back through all of those years and put some perspective on the way in which the PC, and Microsoft in particular, have come to dominate our electronic world.

Back in the eighties it was all up for grabs. The PC was a bulky device which ran DOS in one or two different flavours. Every time you wanted to go from a Spreadsheet to a Word Processor you had to reboot and there was no direct way to interface with the machine without writing a a script file or loading QBASIC writing a BASIC program and then running it.

It was also not the only kid on the block. There was the Dragon, The Commodore, The Atari and the BBC B all of which were more than capable of giving the PC a run for its money.

So where did it all go wrong? Maybe the botched launch of the QL had a lot to do with it since the word got around it was not to be trusted as a system. Maybe Sinclair's increasing financial problems added to that air of uncertainty. Certainly the handling of the QL market by Amstrad (a company whose name was taken from Alan M. Sugar Trading - not too far removed from the other great eighties icon of Trotters Independent Trading - co-incidence maybe?) did little to help the fortunes of the QL.

My Back Pages

These ruminations were further compounded when I went to pick up a lot of QL and Spectrum stuff from an ex user. Looking back at the vast amount of stuff that was available to us in the early days of the QL it does seem surprising that we did not do better. Some of you may point to the introduction of the early 16 colour system or GUI as being points at which the ascendancy of the PC began to make itself felt. Even after that Amstrad had finally put the knife into further QL development we were still doing well. As I have said many times I much preferred working with my QL to my girlfriend's (now wife's) XT machine for all of it capacity.

1 Billion Tears PC

Gartner Dataquest also brought out a report this month stating that sales of PC passed the 1 billion mark in April. This does not mean that there are one million in use, however. As those of us who visit out local council dumps will know there are a large number of older machines languishing there (not to mention the pile outside Dilwyn's window) and there are even more in charity shops and the lofts of people who spent so much on them and cannot bear to throw them away when they become obsolete (about three weeks after you have bought it if you are quick off the mark with the latest stuff).

Gartner saw the early market as being confused by the various different types available all of which had different file systems and, in the case of the QL and the Amstrad CPC machines different media. By doggedly sticking to their own particular formats and refusing to take on board the emerging technology many of these systems sealed their fate.

Had the Internet been around then maybe more would have survived but the sheer exasperation of trying to give a simple text file to someone who used a different system led many people to abandon their computers in favour of the strong IBM/PC model which was being sold in increasing numbers.

In the end, I suppose, we lost out to better marketing, a compatibility issue and more concerted programming.

Strange how all of this coincided over a couple of days though.

Back in the USA

This year's US show was a very upbeat experience for QL users I thought. It was good to see that people were working on new developments in both the hardware and software environments.

Jim Hunkin's demonstration of QDT was something I found particularly uplifting. He is really the first person to start to use the available colours in some real programming and the results are very good. Even in it's partly working alpha state the sheer flexibility and potential of the ideas shone through.

During the discussion of the system which followed his demonstration he made a lot of notes about the suggestions raised by the people who were watching so it is clear that he has the end user very much in mind.

The concepts may not be to everyone's liking but they do bring the QL more into the graphical area occupied by many of the more modern computer systems. There are still those for whom the mouse is the work of the devil and the long command lines typed into the interpreter are the holy catechisms or mantras. Me, I am all for point and click, get the work done.

Doing it the Hardware way

You may be forgiven (only just) if you feel I have something against hardware. In the many and lurid outpourings on the QL Users list I have been seen as being somewhat inclined towards the use of QPC2 than towards a hardware solution.

The truth is that I have just as much enthusiasm for both sides of the fence. I love the portability that QPC2 has given me and the fact that I can have a PC in the same box as a QL. I have also, over the years, enjoyed messing around with various bits of QL hardware. The whole 'Electronic Meccano' feel gives you a different feeling to that of the standard PC 'plug it in and load the drivers' (more on this a little later).

It was, therefore, good to see Nasta demonstrating, at the US Show, his enthusiasm for building and designing new hardware for the QL. His concepts of how the QL hardware scene should progress are, as usual, carefully thought out and neatly executed whilst,at the same time, being quite visionary in some areas.

His next new project, which would seem to be well advanced, is the Super IDE board. I am sure there will be other mentions and descriptions of this in the magazine so I will keep my contribution short but, given the thoughts expressed above about compatibility, there are a couple of considerations.

The new board will have several normal IDE channels as well as the facility to address Compact Flash cards. For those of you who do not know these devices they are used in many cameras and handheld PDAs as removable storage devices.

The question I should have asked Nasta but did not think of at the time is what kind of file system is envisaged for the interface?

File Under.....

At the moment we have three systems in current use.

The old Qubide is based on the 'Rebel' interface and heavily modified by Phil Borman. This had quite a few good points to it but there were also some drawbacks in instabilities in the way it stored its file map. On the whole it has been a stable system for most people and it was usually the fragility of the hard drive that caused problems rather than the interface or driver.

QXLs and QPC both use a different method for the hard drive. The base unit is a PC formatted drive which is the same one used for the PC's operation. SMSQ will then create a file on the PC's hard drive. Nominally this is called QXL.WIN and seen by the PC's system as such. You can rename this in QPC but the QXL never had that facility. Once the emulation is started it will see the QXL.WIN file as WIN1_. There are facilities in the configuration of QPC2 to create several of these files on the same media and give them names. Up to eight at a time can be used by SMSQ/E. I will not go too deeply into this now but the system becomes very useful for CD access.

The Q40/60 uses an Atari format for the hard drive and comes with a partitioning program which can be used to create different 'virtual drives' on the available space. the original system would only allow four partitions to be created although this may have changed since. I have not seen any discussion of this anywhere so maybe someone can enlighten us.

The thing the three systems have in common is that they are mutually incompatible and that is, in my opinion, a major sticking point. Now I admit that it would be hard to change tack for all of these systems at this late stage but we desperately need to try to build in some form of new compatibility into our new hardware. Although it may seem a bit like bias on my part I would like to see that compatibility built around the QXL.WIN format and I have compelling reasons for saying that.

School Uni-information

The thing I would like Nasta to consider is to have the QXL.WIN file format for the new IDE device. This would mean that older users would have to change their data over from the old system to the new one but it would only be the container that would change. The internal structure is the same whether it is mounted on a Qubide, a Q40 or QPC. If you can build QXL.WIN support into the IDE device then you can transfer data from device to device via compact flash cards with complete ease. Given the size of the QL's data and program files you can easily fit a complex system onto 128Mb so you could move your whole system around with ease across all of the platforms.

Now I know that some of you are going to pop up and talk about QXLTools and some of the other programs available to read these files and I agree that the facility is available now but it is not a part of the system. Many of you who read this magazine will have heard me say this before but I will say this again. The drawback with all of these extensions and add ons is that they do require some tinkering and setting up to be usable and for some people that is just too much for them to do. At the very least reading, copying, deleting and all of the usual file operation should be available without a complex setup system. Our community is not just a bunch of people who are all technically advanced and completely literate in programming. It is also ordinary people who want to swap files with other people. SMSQ/E does allow the writing of PC format disks and we should really consider extending this system to bigger devices.

Ethernet or not

Another welcome innovation that will be on the new Super IDE is the optional Ethernet port. This will not mean instant networking between these devices and other computers or even between Super IDE equipped QLs because we will still need some driver software but it is a step in the right direction. We also need some cards and drivers for the Qxx series so we can connect both of these systems to standard PC systems and we have a great deal more flexibility.

New project anyone?

Ethernet is very useful on my PCs at home and at work. I plug my laptop into the network at home and then map the drives. I then configure the device list in QPC2 to have the locate the QXL.WIN files on the newly mapped drives and have them on the list. That way I can access the files on either machine just as if they were on the one I am using. It also means I can backup the files between the two very easily. I would love to be able to do this on the Q40 and the MinsisQL at the same speed and on the same network.

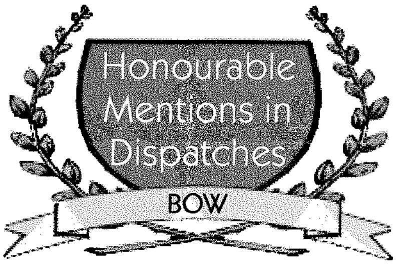
Fly Like a QPCagle

Talking of QPC2, Marcel has added more improvements to the code over the last two months. He has done what many people have asked him to and changed the slave block handling so that files load faster. This speeds up the operation of the program considerably and makes the whole thing very slick to use. He does hope to be able to add similar changes to the next release of SMSQ/E for the other systems too. This release of QPC2 also sees the first steps towards adding Window Manager support for the new colours and that will enable programmers to adapt PE programs to use the extended palette.

The current version of QPC2 did throw up one problem at the recent Eindhoven show. My Invoicing program is compiled with QLiberator and I needed to make a few changes to it at the start of the show. When I recompiled it the whole system crashed. Wolfgang Urlich, who was there at the time and who had also updated his version of QPC2, tried the same thing with one of his program and experienced the same problem. Jochen, however, was able to compile programs with ease. I loaded the previous version of QPC2 and the file compiled perfectly.

I have been looking into this and I have come to the conclusion that the difference was that both Wolfgang and I were compiling programs created with EasyPtr and both of us were adding menus to the programs. Jochens, however, were straight BASIC listings without attempting to attach other files. This will take some investigation but I suspect that the fault lies in the way the QLiberator works.

It is not unknown for QLiberator to have problems with more modern systems and it is not strange that changes to the way that the operating system handles things like the memory should throw up odd problems. I hope that we can solve this one because the speed gain in the current version is so noticeable. In the meantime I suggest that, since this would seem to be the only problem with the current version, users who upgrade keep a copy of the previous version in case they need to compile something. Anyone who has any thoughts on this subject should contact Marcel so he can put the problem to bed.



Honourable Mentions in Despatches

This is a leftover from last issue's column. In my haste to get the copy to Jochen in time for the US show I managed to leave this out. So, sorry for the lateness but this is too good to go without a mention.

All the QColours of the Rainbow

We have had a great number of 'firsts' recently but this is the first QL program that can seriously emulate the Windoze ability to use skins and that, in itself, is worthy of some great trumpet playing. Combine this with the fact the the thing is so incredibly useful and you have a great little utility all for free.

The First part of this program came from Wolfgang Urlich who demonstrated it to me at an Eindhoven show early in the year. I wanted to get a deeper red colour on a program I was playing with and the colour palette did not really offer that. Wolfgang gave me a small utility to allow me to find the colour I wanted and then use the 16 bit colour number to get it into the program. You can select either colour, hue or saturation and move the sliders around until you arrive at the colour of your choice. Once you have done this you click on the little button on the bottom and the resulting hex code is copied to the stuffer buffer. A quick ALT-SPACE in QD and there you are.

All Wolfganged up

If this was not enough of good thing Wolfgang Lenerz got in on the act and added his own little refinement. He had been working on a utility to allow people to use a bitmap as a background in an EASYptr menu. When you combine the two things you get a stunning effect on a very useful little program.

You can grab this little program from most of the usual free sources including Thierry Godefroy's and Dilwyn's website and it is well worth the effort of getting a copy, even if you do not do any programming, just to see the effects. It will, of course, only work on the systems which use the extended colour drivers but that is an increasing number of people now.

This is a really good piece of work - well done the Wolf Gangs!

And Finally....

This is the time of year that many of you head off to the coast for a bit of sun and surf. Most of the newspapers have recommended reading lists. Here are a couple from me...

The C Handbook

Pier To Pier Networking Made Easy

Have a good summer holiday.

The QL Show Agenda

2nd Irish QL Show - Ireland

24th to 26th August

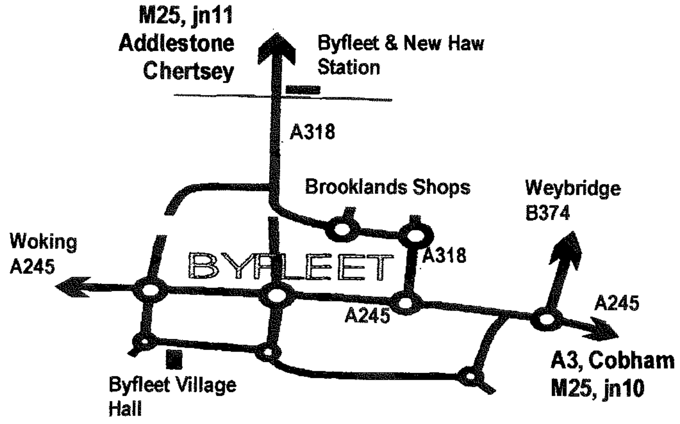
Exact venue to be announced - best contact Darren Brangh, the organiser (see News section).

Quanta workshop - (GB) Byfleet

Sunday, 22nd of September, 10am - 4pm

The Byfleet Village Hall.

The Hall is just inside M25, between jns 10 & 11, just South of A245.
From jn11 go towards Weybridge (East) then turn right (South) onto A318 towards Brooklands, then through the old track onto A245. Right, then left at second roundabout. Left at little roundabout and Hall is on right.
From jn 10, A3 towards London, left onto A245 at next junction, towards Woking. After A318 joins, turn left at second roundabout – see above.
Free Parking, and all the usual attractions. If that isn't enough, Brooklands Museum (aircraft and motor racing) is just up the B374: or there's a Bus Collection on the A245 to Cobham: or I'll explain how to drive to the RHS Gardens at Wisley.
The station is on the Waterloo – Woking line.



Contact: Ken Bain, 01932 347 432 **kenb@bcs.org.uk**

German QL Show - (D) Berchtesgaden

Two-Day Event, Sat./Sun, 5th/6th October

Hotel Schwabenwirt, Koenigsseer Str. 1, D-83471 Berchtesgaden

Driving directions

1. From München / Munich international Airport

a) Take the Bus or S-Bahn (railroad) that leads you in 45 min. to Munich Hauptbahnhof (railwaystation) and from there you get the ticket to Berchtesgaden. Maybe you will have to change the train at Freilassing. You will arrive there in about 4 hours. Ask for special prices (groups, weekend).

b) rent a car and get to the autobahn A8 Muenchen-Salzburg, after 150 km you will cross the border to Austria (you will need a vignette for the austrian autobahn, buy it at the last petrol-station in germany), in austria take the direction Villach/Klagenfurt/I/Yu (not Salzburg or Wien/ Vienna) then leave the autobahn at the exit Salzburg Süd/Berchtesgaden, there turn right and follow the street straight up to Berchtesgaden (about 16 km). You never turn left or right and find the big bus-/railway-station in the center of BGD at a traffic-light. Turn left, cross the bridge and see after 50m the yellow hotel and the parking lot at the left side.

c) as above, but without vignette, slower, but cheaper and over the mountains: Leave the autobahn 145 km after Munich at the exit Bad Reichenhall, turn right at the roundabout and then follow the signs about 22 km along the street B20 around Bad Reichenhall and to Berchtesgaden. Do always take the main-road! There in BGD you will stop the first time at a traffic light near the railwaystation. At the second trafficlight turn right over the bridge and after 50m turn left. That's it.

d) If the weather is fine, you have a lot of time and you want to enjoy the nice alps, leave the Autobahn at the exit Bernau at the big lake Chiemsee (take a trip with the ship to the famous castle on the island!), follow the street B 305 -› Grassau -› Reit im Winkl (good italian icecream!) -› Ruhpolding -› Inzell -› Ramsau -› Berchtesgaden. Arrived there, look at (c). It's a marvellous sightseeingtour along the alps!!!

2. Salzburg Airport:

Take a bus or a taxi to the central station (Hauptbahnhof), from there take the RVO-Bus (the fastest) to Berchtesgaden Hauptbahnhof. (The trip from there to BGD by train via Freilassing lasts very long!) There cross the street and the bridge, after 50m on the left side you will see the yellow hotel. Easy?